# računari 43

oktobar 1988.

časopis za popularizaciju
informatike i računarstva
cena 2000 dinara

**naš stav**
pirati?
ne, hvala!

**komentar**
u kom mikru
leži zec

**intervju**
milion osmeha
čede mraza

**naš test**
diamond
optasm
smartwork
grasp
perfect pal
beebdos

**hardver**
keš memorije
ploteri
akceleratori






# NOVKABEL ELEKTRONSKI RAČUNARI – NOVI SAD

## PROIZVODNI PROGRAM ELEKTRONSKIH RAČUNARA

**— RAČUNARSKI SISTEM ET-188A**
Personalni računar na bazi INTEL-ovog mikroprocesora 80188 operativni sistem MS DOS 512 KB RAM memorije disk drajv 360 KB fiksni disk 20 MB paralelni i serijski interfejs (monitor LAN/lokalna mreža)

**— POSLOVNI SISTEM MPS-4**
Izrađen na bazi mikroprocesora Z80 operativni sistem MP/M CP/M 320 KB RAM memorije fiksni disk 86 MB strimer traka 40 MB 4 terminala linijski i matrični štampač (mogućnost proširenja do 24 terminala)

**— RACON**
Računar koncentrator na bazi INTEL-ovog mikroprocesora 80286 operativni sistem XENIX 4 MB memorije do 4 disk drajva od 360 KB 2 fiksna diska od 86 MB strimer traka 40 MB konzole i 8 terminala (mogućnost proširenja do 24 terminala)

**— TERMINALI IZ SERIJE PT-100**
IBM kompatibilni terminali oblikovani prema specifičnosti poslova koje korisnik pomoću njih želi da obavlja

**— APLIKATIVNI SOFTVER**
Industrija turizam administracija, trgovina poljoprivreda prosveta

**— PROCESNO UPRAVLJANJE**

**— OBUKA KADROVA I KONSULTANTSKE USLUGE**

**— SERVISIRANJE OPREME**

**— ORGANIZACIJA SVIH VIDOVA KURSEVA IZ OBLASTI INFORMATIKE**

## RACON — računar koncentrator



*Nova tehnologija je rezultat zajedničke saradnje instituta „Boris Kidrič" iz Vinče i inženjera Novkabela*

*Računarski sistem spada u klasu mini-kompjutera Višeprocesorska mašina je zasnovana na savremenom mikroprocesoru „INTEL 80286" Radna memorija u najjačoj verziji ima kapacitet od 4 MB a računar može da podrži rad 16 korisnika istovremeno kao i da se uključi u računarsku mrežu sa većim ili manjim kompjuterima*

*Po svojoj nameni ovaj računarski koncentrator je univerzalnog tipa te se uspešno koristi u obradi poslovnih naučno tehničkih procesnih razvojnih informacija u procesima prikupljanja i prenosa podataka itd*

*Sve kartice za ovu mašinu proizvedene su u novosadskoj fabrici kablova kao i kompletna mehanika a uvozne komponente su jedino čipovi i memorija*

*Mašina radi i pod standardnim operativnim sistemom XENIX pripada priznatom svetskom standardu te može da ima široku primenu*

*Obzirom na veoma efikasno organizovanje LAN-ova i protok informacija kroz čvorove transfer prema drugim računarskim sistemima neosporno je da ova 16 bitna mašina ima visoke hardverske i softverske performanse i podržava svetske trendova u ovoj oblasti*

*Ako ovome dodamo mogućnost podrške personalnih kompjutera te da korisnik može razvijati programe u ASM COBOL-u i Fortranu paskalu i C-u njegova univerzalnost dolazi još više do izražaja*

## PROIZVODNI PROGRAM NOVKABEL — OLIVETTI

**— M-240**
Personalni profesionalni mikroračunar na bazi INTEL-ovog mikroprocesora 8086 na 10 MHz 640 KB RAM-a disk drajv od 360 KB 20 MB fiksni disk 20 MB strimer traka

**— M-280**
Personalni profesionalni računar AT varijanta na bazi INTEL-ovog mikroprocesora 80286 na 12 MHz jedno ili višekorisnički do 4 terminala u vezi 1 MB RAM-a 1 2 MB disketa 5 25 40 MB fiksni disk strimer traka 60 MB MS DOS i XENIX operativni sistemi

**— M-380**
Profesionalni personalni računar na bazi 32-bitnog INTEL-ovog mikroprocesora 80386 brzine 20 MHz 4 MB RAM-a sa mogućnošću proširenja do 52 MB 122 MB disk drajv formata 3 5 135 MB fiksni disk 60 ili 125 MB strimer traka, multiuser multitasking konfiguracija pod XENIX operativnim sistemom

**— LSX 3020/30/40**
Računari zasnovani na 32-bitnom mikroprocesoru MOTOROLA MC 68020 brzina 16 MHz 4 MB RAM-a sa mogućnošću proširenje do 14 MB 1 MB flopi disk strimer traka 40/80 mogućnost proširenja do 64 terminala i fiksni disk 70—1260 MB

— IBM PC kompatibilni štampači
— DM-292 brzina 240 cps
— DM-400 brzina 400 cps
— DM-296 brzina 220 cps/kolor
— PG-208 laserski štampač

**INFORMACIJE RO „NOVKABEL", POSLOVNICA PRODAJE ERA, Tel 021/337-255, 338-199 (lokal 2211)**

**RAČUNARI NOVKABEL**

# računari 43

časopis za popularizaciju informatike i računarstva
izlazi jednaput mesečno
YUISSN 0352-7271
Izdaje BIGZ
oktobar 1988 cena 2000 dinara

## Sadržaj

4) Šta ima novo
7) Komentar
**U kom mikru leži zec**
8) Hardver
**Septembar u Minhenu**
8) Hardver
**Dobro došli u svet robotike**
10) Intervju
**Hiljadu osmeha Čeda Mraza**
13) Hardver/Memorije
**Brzo, brže . . . keš**
17) Periferijska oprema/ploteri
**Ploteri pišu perom**
20) Hardver/akceleratori
**Kako frizirati PC**
22) Komercijalni softver/programski jezici/Workfort
**Fort bez Isaima**
24) Komercijalni softver/integrisani paketi/Diamond
**Dijamantski rez**
28) Komercijalni softver/asembleri/Optasm
**U dosluhu s đavolom**
28) Komercijalni softver/CAD
**smARTWORK**
31) Komercijalni softver/grafika
**Graap**
32) Komercijalni softver/korisne rutine/PerfectPal
**Savršena avaštara**
34) Komercijalni softver/korisne rutine/BeebDos
**Ne okreći se sine**
35) Javni softver
**Sinišin gambit**
36) Softverski podsetnik/MCoder II
**Poslednji Mohikanac**
38) Izbor knjiga
40) Adaptacije/yukeyboard sys
**Reprogramiranje tastature**
41) Operativni sistemi/TOS
**Bube u mišu**
42) Operativni sistemi/VMS
**Editori teksta**
43) Tehnike programiranja/inteligentno pretraživanje
**Ni krstić ni kružić**
45) Tehnike programiranja/„komodor"
**Na grafičkom poligonu**
46) Tehnike programiranja/Z80/asembleri
**Tokeni u ringu**
48) Algoritmi
**Potkresivanje stabla**
48) Algoritmi
**Potkresivanje stabla**
50) Help
52) Dejanove pitalice
53) Bajtovi lične prirode
56) Samogradnja „time 011"
58) Pet plus
61) Razbarušeni sprajtovi



**Uz naslovnu stranu**

**Svojim završenim programom za pretraživanje, Bratislav Đorđević nam je za oproštaj za letom prozašao jedna vrsna programerku. To je Bojana Ostojić, učenica srednje škole matematičkog smera iz Pereča. Objektivom ju je ovekovečio famozni Darko Tadić, ovoga puta bez uobičajnih programerskih alatki.**

U sledećem broju nova akcija

## Mi vama diskete, vi nama utiske!

Izvestan broj čitalaca „Računara" dobiće na test besplatno po dve „Magmedia" diskete uz jedinu obavezu da nam pošalje nekoliko rečenica o svojim utiscima o njihovom kvalitetu i pouzdanosti

**Izdaje**
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

**Generalni direktor**
Dobrosav Petrović

**Zamenik generalnog direktora**
Antun Martić

**Glavni i odgovorni urednik časopisa „Galaksija"**
Stanko Stojiljković

**Glavni i odgovorni urednik**
Jovan Regasek

**Stručna redakcija**
Žarko Berberski, Voja Gašić, Slobodan Perović, Dejan Ristanović
Jovan Skuljan, prof. dr Dušan Slavić
Nevenka Spalević, Zoran Životić, Anđelko Zgorelac

723

**Sekretar redakcije**
Zorka Simović

**Tehnički urednik**
Dušan Mijatović

**Pomoćnik tehničkog urednika**
Mirko Popov

**Marketing**
Sergije Marčenko

**Stalni saradnici**
Neda Aleksić, Slobodanka Ast, Žarko Berberski, Viktor Carovski, Zoran Cvijetić, Ninoslav Čebrić, Voja Gašić, Željko Jurić, Blažimir Miše, Zoran Obradović, Slobodan Perović, Miodrag Polkonjak, Aleksandar Radovanović, Dejan Ristanović, Jelena Rupnik, Duško Savić, Dušan Slavić, Jovan Skuljan, Nevenka Spalević, Vlada Stojiljković, Saša Svilica, Milan Tadić, Žarko Vukosavljević, Anđelko Zgorelac, Zoran Životić

**Izdavački savet „Galaksije"**
Dr Rudi Dabijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Glišić, Esad Jakupović, Velizar Mesiać, Nikola Pejić, Željko Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vasović, Milivoje Vuković

**Adresa redakcije**
11000 Beograd, Bulevar vojvode Mišića 17/II

**Telefoni**
650-181 (sekretarijat)
653-748 (redakcija)
650-528 (prodaja)
651-793 (propaganda)

**Rukopisi se ne vraćaju**

**Štampa**
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar vojvode Mišića 17

**Pretplata u zemlji**
Za jednu godinu 24 000 -
Za šest meseci 12 000 -

Na žiro račun RO BIGZ 60802-603-23254

**Pretplata za inostranstvo**
Za jednu godinu 48 000 -
Za šest meseci 24 000 -

**Inostranstvo u devizama**
Za jednu godinu 15 USD, 34 DM, 28 CHF, 10 GBP i 114 FRF
Za šest meseci polovina od navedenog iznosa
Na žiro račun RO BIGZ 60811-620-16101-820701-898-03377
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 28/72, ovo izdanje oslobođeno je poreza na promet



# Šta ima novo

Komercijalni softver

## Stono izdavaštvo na komodoru

Zašto bi stono izdavaštvo bilo rezervisano za vlasnike PC-ja? Program *PaperClip Publisher* firme „*Electronic Arts*" (američki telefon (415)571-7171) košta samo 50 dolara a ipak omogućava „komodoru" 64 i 128 da obrađuje tekstove duge čitavih 50 strana uz kombinovanje teksta i grafike. Program je posebno prilagođen izdavanju novina, pošto omogućava rad sa kolonama teksta a grafički editor uračunat u cenu prilagođava crteže urađene uz pomoć raznih grafičkih programa formatu *PaperClip Publisher*-a. Minimalni hardverski zahtevi za upotrebu programa uključuju 1541 ili 1571 disk i miš odnosno džojstik.

Interfejsi

## PC — APPLE — PC



*WPC Bridge* firme „*Cordata Technologies*" (američki telefon (213)603-2801) zapravo predstavlja čudesnu kombinaciju dve totalno nekompatibilne računara u istoj kutiji — IBM PC i Apple 2. *WPC Bridge* naravno sadrži mikroprocesore 8088 (4.77 MHz) i 65C02 (1 MHz), 512 kilobajta RAM-a, koji se mogu proširiti do 768 K, i grafički interfejs rezolucije 640*400 odnosno kada se imitira Apple, 720*350.

Cena od 1895 dolara uključuje dve interne disk jedinice od 360 K za PC i dve Apple II kompatibilne ekspanzione slote, AT tastaturu i monitor. Zar kombinacija nije zanimljiva?

Štampači

## Star Multifont LC24-10

*Star Multifont LC24-10* možda nije brz kao *Honeywell Bull 4/68*, ali zato košta svega 380 funti. Radi se od 24-pinskom štampaču koji u svakom sekundu ispisuje 142 standardna ili 47 NLQ slova u jednom od četiri ugrađena fonta (*Courier*, *Prestige*, *Script* i *Orator*). Korisnicima kojima ovi fontovi nisu dovoljni na raspolaganju su specijalni kartridži. Što se veličina slova tiče, u svakom fontu je na raspolaganju *pica* (10 cpi), *elite* (12 cpi), *semicondensed* (15 cpi) i *condensed pica* (17 cpi) i *condensed elite* (20 cpi). Poput većina Star-ovih štampača LC24-10 može da ispisuje slova koja su višestruko šira i viša od standardnih. Britanski telefon firme „*Star Micronics*" je (01)840 1800.

Veliki sistemi

## RISC supermikro



*pKS FALCON* je stoni supermikrokompjuter koji u svakom sekundu izvršava 33 miliona instrukcija. Za ovu brzinu zaslužan je prvenstveno paralelni RISC procesor koji radi na 30 MHz, I/O procesor 80C186 i Ethernet koprocesor 82586. RAM od 4—16 megabajta i hard diskovi od 80 do 760 megabajta dopunjavaju ovu konfiguraciju snova koja košta 15 do 25 hiljada funti i koju biste (zar ne?) svakako kupili da izvoz iz Britanije nije zabranjen!

Mikroprocesori

## 68040 dolazi

Na nedavnoj konferenciji za štampu održanoj u Palo Altou „Motorolin" predstavnik Mari *(Marray) Goldman* je potvrdio da je razvoj mikroprocesora 68040 priveden kraju, ali se nije izjasnio o performansama. Goldman je takođe odbacio razne spekulacije o datumu izlaska na tržište, tvrdeći da on je još nije poznat i da zavisi od mnogih parametara. Pravi političar!

Profesionalni hardver

## Brza Furijeova transformacija



Svaki inženjer dobro zna koliko je Furijeova *(Fourier)* transformacija značajna za razne oblasti tehnike. Zbog toga su razvijeni mnogobrojni algoritmi za brzu Furijeovu transformaciju kojima ponekad dobro dođe i malo hardvera. FTT procesor firme „*Protek*" (britanski telefon (01)245 6844) pomaže „*Hewlett-Packardovim*" računarima iz serije 200 i 300 da procesiraju signale fantastičnom brzinom od 100 tačaka u milisekundi. Prepreka može da bude jedino cena od 2000 dolara.

Mreže

## At sa trideset terminala

Ko kaže da je za višekorisnička sisteme neophodan 32-bitni mikroprocesor? Obratite se firmi „*UltraTek*" (američki telefon (415)364-1080) i dobićete konektore i kablove koji omogućavaju povezivanje trideset ASCII terminala na serijski port bilo kog AT klona. U cenu od 700 dolara uračunato je pet kartica koje treba ugraditi u AT-a da bi on kontrolisao pet terminala i operativni sistem *Alloy*. ako želite da se približite maksimalnoj specifikaciji, to jest svojevrsnoj terminalskoj učionici, moraćete da platite još poneki dolar za adapter i nove kartice.

Štampači

## Sve bolji laseri

Budućnost štampača je laserska tehnologija zasnovana na boljim mašinama, izjavio je nedavno Luis Vilalobos *(Villalobos)*, predsednik poznate korporacije „*Conografic*" koja proizvodi kartice pomoću kojih klasični laseri „progovaraju" PostScript. Vilalobos je takođe izjavio da će se na tržištu uskoro naći jeftini laserski štampači koji ispisuju 600 tačaka na svakom inču i da će ih pratiti pad cene kolor laserskih printera. Za kraj 1989. Vilalobos predviđa pojavu laserskih štampača koji će ugroziti sadašnji fotoslog — 1000 do 3000 tačaka po inču! „Računari" da očito znali o čemu da pišu!

Računari u izlogu

## Igračkica od megabajta



Igračka sa slike je igračka samo po dimenzijama (11*38 cm) — radi se zapravo o 80386 mašini koja zajedno sa 512 K RAM-a (memorija se proširuje do 5 megabajta), disk jedinicom od 3.5 inča i hard diskom od 30 megabajta košta 3000 dolara. Firma „*Advanced Logic*" (američki telefon (714)581-677) prodaje i AT klon identičnih dimenzija, cena ne prelazi 2000 dolara.

Štampači

## Iglice se ne predaju

Matrični kolor štampači se ne predaju pred naoko znatno atraktivnijom laserskom tehnologijom: upravo lansirani matrični štampač firme „*Honeywell Bull*" (američki telefon (415)974-4340) u svakom sekundu ispisuje 600 karaktera, što je, čak i kada se uzme u obzir činjenica da su ovakve specifikacije uvek pomalo preterane, brže od većina lasera. Novi štampač kome je dodeljena oznaka 4/68 može naravno, da radi i u LQ modu u kom slučaju ispisuje 150 znakova u sekundi. Glava printera je 18-pinska a tajna fantastične brzine je mogućnost da se dva reda iglica pri *draft* štampanju razdvoje i tako odjednom ispišu dve kolone tačkica. Što se cena tiče valjda niste očekivali nešto jeftino — 2450 dolara!

724

Mikroprocesori

## Za prave matematičare



Koprocesorska kartica sa slike omogućava čak i najsporijem XT-u da u svakoj sekundi izvrši 8—25 miliona operacija sa racionalnim brojevima, što predstavlja dosad neprevaziđenu cenu od stotinak dolara po MFLOPS-u. Osnova kartica je AT&T-ov procesor DSP32 koji, opremljen sa 32 kilobajta statičkog RAM-a, preuzima sve funkcije „Intelovog" 8087 odnosno 80287. Sama kartica košta 745 dolara, dok se za 995 dolara dobija i C kompajler sa potrebnim bibliotekama. Američki telefon proizvođača, firme *„Communications Automation & Control"* (215)776-6669)

Štampači

## Trake koje se ne troše

Trake koje se ne troše? Pa, ne baš da se ne troše, ali traju 20 puta duže od standardnih traka za Epson FX i RX, Apple Imagewriter i druge popularne matrične štampače. Nevolja je samo što trake koštaju bar deset puta više od standardnih: 50 dolara za Epson verziju. Možda ipak poželite da popričate sa stručnjacima firme *„Chronos Computers"* — američki telefon je (818)455-8200)

Štampači

## Postscript za laserjet

Vlasnici popularnog *„Hewlett-Packardovog"* laserskog štampača *Laser Jet II* odnedavno mogu da uživaju u blagodetima PostScripta zahvaljujući *Cono-Desk 6000* kontroleru firme *„Conographic"* (američki telefon (714)474-1188). Proizvodnja tvrdi da je simulacija bar 10 puta brža od najbržeg konkurenta i da su u „sitnu" cenu od 3000 dolara uračunati brojni PostScript fontovi za Venturu i PageMaker

Kalkulatori

## „Šarp" ili . . .

Kalkulator pomenut na slici bi možda bio vest za „Galaksiju" iz 1981, ali za „Računare" iz 1988. Ne bi bio vest da je „Šarpov" ali — pogledajte malo bolje ime pirata i piratal

Komercijalni softver

## Rečnik na disku

Ne radi se o uobičajenom rečniku za proveru spelovanja nego o pravom engleskom rečniku koji omogućava da proverite značenje svake od 80 000 unetih reči. Program se integriše sa bilo kojim popularnim tekst procesorom i omogućava pronalaženje reči za svega 2—5 sekundi na AT kompatibilnom računaru.

Osnova za ovo dvomegabajtno kompjutersko remek-delo firme *„Proximity Software"* (američki telefon (305)566—3511) je *„Merriam Webster"* rečnik modernog engleskog. Cena je sasvim prihvatljiva — 100 dolara

Naličje računarstva

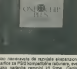
## Pretnje novih tehnologija

Redovni elementi raznih reklama za laserske štampače i skanere je digitalizovana ili fotokopirana novčanica od jednog dolara ili hiljadu dolara. ... neko se čak dosetio da Džordžu Vašingtonu docrta brkove! Mogućnosti nove tehnologije koja je idealno oruđe za falsifikatore zabrinuli su američke federalne vlasti koje su odlučile da predizajniraju dolar. Novi dolar će ući u upotrebu tokom 1990. godine i sadržaće specijalno poliestersko vlakno, dok će slova USA biti izdignuta nekoliko milimetara u odnosu na površinu papira.

Nas, na sreću, ne muče slični problemi — dok bi neko osvojio tehnologiju laserskog umnožavanja neke novčanice, one bi vredele manje od papira na kome se štampa!

Periferijska oprema

## Mikro kanal na čipu




Ako nameravate da razvijate ekspanzione kartice za PS/2 kompatibilne računare, svakako nabavite razvojni kit firme *„Capital Equipment"* (američki telefon (617)273—1818). U cenu od oko 1000 dolara uračunata je osnovna kartica sa kompletnim hardverom potrebnim za dekodiranje i logičke operacije, integrisanim na jednom čipu. Prateći softver obuhvata LIM/EMS drajvere i programe za analiziranje ID oznaka koje mikro kanal dodeljuje. Ukoliko se vaše kartice docnije budu prodavale, svaki primerak „superčipa" koštaće po 35 dolara

Masovna memorija

## Za zdrave hard diskove

Solidna predohrana protiv različitih virusa i „trojanskih konja" je zabrana upisa na hard disk. Kada se već proizvođači ovih naprava nisu dosetili ničem sličnom, firma *„S & S Enterprises"* (britanski telefon (0494)724-201) za svega 10 funti prodaje neku vrstu vakcine koja, jednom aktivirana, onemogućava sve upise na disk. Softverska zabrana, jasno, nikada nije neprobojna, ali postoji nada da ni autori virusa nisu trošili pare na sve moguće vakcine!

Periferijske opreme

## Džepni modem

Posle brojnih džepnih PC-ja evo i jednog džepnog modema. *„Pocket Stradcom"* firme *„Dateflex Design"* (britanski telefon (01)543-6417) dug je 75, širok 55 i visok 25 milimetara i napaja se iz alkalne baterije od 9 volti, koja obezbeđuje desetak časova „razgovora" sa nekom kompjuterskom mrežom uz brzinu komunikacije od 300 ili 1200 bauda. U cenu od 200 funti uračunat je ispravljač koji ćete koristiti kada vaš džepni modem nije na terenu

Programski jezici

## Novi Borlandov prolog



„Borlandov" turbo prolog 2.0 predstavlja priličan korak unapred: baze podataka sada može da budu eksterne a korisnički interfejs je potpunije prilagođen „Borlandovim" standardima. „Borlandovi" programeri su odgovorili i na optužbu da njihov prolog ne odgovara Edinburškom standardu — u verziju 2.0 ugrađen je predprocesor (pisan na turbo prologu!) koji analizira programe usklađene sa standardima jezika i prilagođava ih Turbo trikovima. Brzina izvršavanja, na žalost, nije bitnije povećana a i cena je ostala — 150 dolara

Operativni sistemi

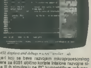
## Unix za mac II

Iako ove godine očito ne planira promociju novog hardvera, firma „Apple" ne zaboravlja svoj „makintoš 2" — početak leta nam je doneo operativni sistem A/UX koji zapravo predstavlja AT&T Unix System 5 Release 2 prilagođen mikroprocesoru 68020 i „makintoševoj" arhitekturi. Pošto je Unix bazično višekorisnički operativni sistem, vlasnici „meka 2" sada mogu da nabave dva terminala i povežu ih serijskim linijama: centralni računar će ih opsluživati podržavajući istovremeno i programera koji sedi za glavnom konzolom — tri radna mesta umesto jednog! Cena paketa od oko 8500 dolara obuhvata i hard disk od 80 megabajta.

Adresa kompanije „Apple" je, za slučaj da je nikada nismo objavili, *20525 Mariani Ave, Cupertino, CA 95014, U.S.A.*

Profesionalni hardver

## 8051 na PC-JU



*SimCASE displays and debugs m...roc...troller ...als*

Inženjeri koji se bave razvojem mikroprocesorskog softvera za 8051 obično koriste Intelove razvojne sisteme ili ih simuliraju na PC kompatibilnim računarima. Ova dvostruka simulacija teško može da bude brzinski šampion pa je firma *Archimedes Software* (američki telefon (415)567-4010) ponudila pravi 8051 simulator za PC-je. U cenu od 885 dolara uračunat je sam simulator, *cross* kompajler za C i veoma moćni debager izvornog koda. Dodatni poklon je *Performance Analyzer* koji precizno određuje vreme izvršavanja svakog segmenta koda dajući tako predstavu ponašanju čitavog sistema u realnom vremenu. (D. R.)

# Pirati? Ne, hvala!

Ako bi jugoslovensku kompjutersku scenu trebalo opisati jednom rečenicom, ta bi rečenica verovatno glasila *preskup hardver, besplatan softver*. Dok se za preskup hardver možemo zahvaliti našoj inflaciji, kursu dolara, nezasitljivim proizvođačima kompjuterske opreme, saveznoj vladi i mnogim drugim faktorima, pirate zaslužne za besplatan softver nije teško imenovati. Softversko piratstvo je dosledno pratilo naš računarski bum: što je neki kompjuter bivao popularniji, to su se programi za njega više piratovali. U samom početku behu mali pirati a onda je, u skladu sa svoga puta dosledno poštovanim ekonomskim zakonitostima, došlo do podele rada, udruživanja i drugih oblika kooperacije i konkurencije: pojavilo se nekoliko velikih pirata - dobavljača softvera, nekoliko velikih distributera čiji promet dopušta iznajmljivanje osrednjih sajamskih štandova i bezbroj malih preprodavaca koji su započinjali karijeru oglasom *Imamo sve ono što i taj-i-taj ali 20 posto jeftinije*.

Pominjanjem malih oglasa dolazimo do doprinosa domaće štampe razvoju piratstva: objavljivanje prvog piratskog oglasa u "Galaksiji" (januar 1983.) prihvatili smo bez ikakvog razmišljanja smatrajući da se radi o jednom od prvih vesnika dugo priželjkivanog kompjuterskog buma; ukoliko smo u tom vesniku i videli nešto loše, verovali smo da će slična pojava ubrzo odumreti. Pokazalo se, na žalost, da ono što je loše ne odumira samo od sebe - broj piratskih oglasa se iz meseca u mesec povećavao pa smo tokom ove godine doživeli da pirati zakupljuju čitave stranice! Pošto je u čovekovoj prirodi da pokušava da opravda svoje postupke tj. da od korisnog pravi ugodno, povremeno smo izražavali ovakvu ili onakvu solidarnost sa piratima tvrdeći da nam dotični pomažu, da su simpatični i duhoviti, da razbijajući programe postaju kompjuterski plemeni, da se bez njih jednostavno ne može. Stalno smo se, sa druge strane, suočavali sa negativnim stavom brojnih saradnika i čitalaca koji su u softverskom piratstvu videli ono što zaista treba videti (kriminal) a u objavljivanju piratskih oglasa ono što oni zaista predstavljaju - saučesništvo. S vremena na vreme javljali su se i protesti protiv nekih "naročito gnusnih" piratskih zločina kao što je umnožavanje igara koje se prodaju u dobrotvorne svrhe. Iako se u sudskoj praksi i kriminalističkim romanima svakom zločinu pripisuju olakšavajuće ili otežavajuće okolnosti, nismo ubeđeni da se treba boriti isključivo protiv piratovanja dobrotvornih programa - zlo je u samoj krađi a ne u krađi ovog ili onog predmeta!

Šta je toliko loše u piratovanju softvera? Ovde ćemo preskočiti sve moralne i pravne dileme koje se svode na otimanje rezultata nečijeg rada i pozabaviti se isključivo praktičnim problemima. Dobro je poznato da je upotreba komercijalnog softvera i dalje prilično komplikovana i da osamdesetak posto kupaca kreativno koristi svega dvadesetak posto potencijala nekog paketa. Kod nas je, prvenstveno zahvaljujući piratstvu, ovaj odnos sveden na 90:10 ili možda 95:5 - najveći deo korisnika je lišen dokumentacije, koristi zastarele verzije koje su uz to i nepotpune ("nebitne" datoteke se usput zature), nedostaje nam bilo kakva pomoć autora softvera koji bi se, da je registrovanih vlasnika više, svakako potrudio oko YU slova i sličnih problema... Neispravan i nepotpun softver računarima polako stvara reputaciju nepouzdanih i preterano komplikovanih naprava od kojih ima više štete nego koristi. Neprijatno.

Argument da se piratovanjem i razbijanjem zaštite unapređuje kompjuterska plemenost je tavode veoma daleko od svake logike. Ako davno uzrečenu tvrdnju da se plemenost na stiče kradom knjiga ostavimo na stranu, čak će i najveći pobornik piratstva priznati da programe "razbija" par ljudi u Jugoslaviji (u poslednje vreme do nas stižu programi "razvaljeni" u Nemačkoj ili Holandiji) dok 99% pirata samo kopira kasete. Ima ih čak i koji su računar zamenili parom duplih kasetofona! Moguće je da je razbijanje zaštita pomoglo određenom broju mladih programera da se uputi u tajne "spektruma" i "komodora 64", ali se na ovakav "razvoj pameti" teško može računati - mnogo je bolje da stimulišemo mlade programere da svoj potencijal usmere u nekom prosperitetnijem pravcu. Ukoliko bi se, najzad, piratstvo svelo u neke razumnije okvire, domaće softversko tržište bi se možda pokrenulo sa mrtve tačke na kojoj već godinama tavori.

Poslednji (bolje bi bilo reći prvi) argument protiv piratstva je da ono ispoljava sve simptome organizovanog kriminala: bande, ucene, podmićivanje i fizički obračuni. Obzirom da se u čitavoj igri ne vrti previše novca, simptomi organizovanog kriminala se ispoljavaju nešto spolje ali su svi tu - setimo se samo jednog od prvih i najvećih pirata koji je poslao izdaka milicionera da "preslišа" prvog klinca koji mu je konkurisao. Softversko piratstvo i šverc su jedini oblici organizovanog kriminala koje štampa na neki način stimuliše; problem šverca je možda prekrupan za nas ali podršku piratstvu svakako možemo da uskratimo. Kada bi, najzad, softversko piratsvo bilo tako dobro kao što neki tvrde, pirati koji su se u međuvremenu "uozbiljili" ne bi toliko pokušavali da prikriju grehe svoje mladosti!

Pošto je većina pomenutih argumenata i zaključaka uglavnom nesporna, još od pokretanja kompjuterskih časopisa se šuškalo o međuredakcijskom dogovoru koji bi ukinuo piratske oglase. Do dogovora, na žalost, nikada nije došlo jer je svaki časopis pronalazio neke interese u manje ili više otvorenoj podršci preprodaji softvera. Što se "Računara" tiče, mali oglasi su doskora uživali reputaciju tiražne kontakt rubrike bez koje bi borbe za čitaoce bile daleko teže. Moramo da priznamo da je opstanku oglasa u značajnoj meri kumovala i naša inercija koja nas je navodila da iz meseca u mesec očekujemo neki dogovor ili neki zakon koji bi stvari rešio umesto nas. Čekanje se, međutim, oteže u nedogled pa je naš časopis odlučio da potpuno samostalno i samoinicijativno ukine piratske oglase. Ukoliko je ova naša odluka preuranjena ili loša, sami ćemo snositi posledice.

Ukidanje malih oglasa ne označava početak nekakvog "krstaškog rata" protiv pirata - piratsvo će i dalje postojati, mi ćemo ga smatrati i komentarisati ali nećemo podržavati čak ni njegove najblaže oblike (ne može se, na primer, izjednačiti društvena opasnost od organizovane preprodaje ukradenog softvera sa aranžmanom u kome dva čoveka zajedno kupe neki paket ili u kome službenik neke firme kupljeni softver koristi i na kućnom PC-ju). Verujemo da se daleko bolje rešenje problema preskupog softvera može naći u kooperaciji sa stranim softverskim kućama koje bi verovatno našle interes u prodaji licenci pojedinim domaćim distributerima koji bi programe prodavali u punim verzijama ali po cenama koje su znatno niže od svetskih - svakako znate da naša Televizija plaća "Dalas", "Dinastiju" i druge svetske hitove znatno manje nego nemačka ili britanska televizija pošto producent zna da mi ne možemo da platimo više i rezonuje da mu je bolje da uzme nešto para nego da ne uzme ništa! Slični aranžmani će, naravno, postati mogući tek kada se domaći distributeri uvere da se bespravno kopiranje softvera koji prodaju neće oglašavati u štampi!

Naša nova orijentacija se ne završava ukidanjem piratskih oglasa - verovatno ste primetili da se "Računari" već par meseci pri prikazivanju komercijalnog softvera služe isključivo originalnim i novim verzijama (*Optasm 1.5* koji prikazujemo u ovom broju je, na primer, izbačen na tržište sredinom avgusta 1988.) koje su propraćene dokumentacijom, podrškom proizvođača i svim sličnim pogodnostima. Verujemo da će aktuelne i kvalitetne informacije koju na ovaj način pružamo umanjiti potrebu za piratovanjem softvera koje je motivisano radoznalošću, nadamo se isto tako da će one radne organizacije i oni pojedinci kojima je program koji prikazujemo zaista potreban naći načine da ga legalno nabave i, prema tome, optimalno primene.

Trudićemo se da prostor koji štedimo ukidanjem piratskih oglasa popunimo vrednijim sadržajima među kojima će se, naravno, nalaziti i mali oglasi u okviru kojih naši čitaoci i saradnici nude originalni softver i hardver, servis, kompjuterske usluge i slične stvari. I ovakvi oglasi imaju svoju tradiciju - samo dva meseca posle prvog piratskog oglasa "Galaksija" je objavila ponudu *Ignaca Uršiča* koji je po narudžbini pisao programe za ZX-81. Oglase poput Ignacovog ubuduće ćemo stimulisati kako cenom tako i pratećim tekstovima trudeći se da naš časopis bude što manje "buvljak" a što više glasilo pravih programera i pravih korisnika računara.

*Redakcija*

U krivom ogledalu

# U kom mikru leži zec

**Računarski listovi, kojih kod nas ima otprilike koliko i takozvanih ženskih revija (ali sa deset puta manjim tiražima!), retko dospevaju u vizira druge štampe. A kada se to ipak dogodi, motivi mogu da budu razočaravajući — barem ako ja suditi po skorašnjem tekstu objavljenom u „Startu" u kome se prave aluzije i na račun „Računara".**

U poslednjem broju zagrebačkog dvonedeljnika „Start" (broj 511 od 20. avgusta 1988), u tekstu „Urednik i njegov mikro" „iz pera" izvesne Sunčane Novak časopis „Računari" pomenut je — na po imenu ali zabune nema — na način koji bi bio uvredljiv da nije neslan i nadasve naivan. Tekst je zapravo posvećen Cirilu Kraševecu, tipu „novog čovjeka" koji je „mlad" uspješan, vozi dobra japanska kola, provjeren je na mnogim kompjuteraškim poslovima" i „bio je jedan od urednika-osnivača prvoga časopisa za kompjutere 'Moj mikro'". Pogodite u kom grmu leži mikro zec!

## Staro u novom

Na sitan „previd" „Startovog" člankopisca da su rodonačelnik domaćih časopisa za kompjutere zapravo „Računari" — prvi broj je izašao januara 1984, a „Moj mikro" je krenuo sa slovenačkim izdanjem juna 1984 a srpskohrvatskim januara 1985 godine — ne bismo se ni osvrtali da u „Startovom" tekstu nema drugih navoda koji izdaleka zaobilaze istinu zlurede su, pa i zlonamerni. Ne mislimo da ovi navodi odslikavaju stav ili nameru „Starta" (sudeći bar po tome što je on o „Računarima" ranije pisao afirmativno) nego pre verujemo da su oni toj reviji poturenı i da su, na žalost, objavljeni bez provere koja prilikom udaranja packi drugim listovima može da spreči ogrešenje o elementarne činjenice i čaršijski red. Možda, da ostanemo do kraja galantni, na treba posebno kriviti ni autora Sunčanu Novak koja je, recimo, jednostavno primila zdravo za gotovo neke „stare stvari" od nekog „novog čovjeka".

Sa stanovišta „Računara" sporan je zapravo samo jedan pasus, onaj isti koji sadrži i neistinu o prvom kompjuterskom časopisu u našoj zemlji:

„(...) i upikos ponekad komičnim prijevodima i korektur tekstova, to (srpskohrvatsko izdanje 'Mog mikra' — prim. E. J.) je ostao časopis koj sve do danas **nisu uspjeli dostići** <1> konkurentski **beogradski izdavači kompjuterske beletristike** <2> (**a jako su se trudili** <3>), naime u sud časopisa 'Moj mikro' vjeruje osjetljivi i sve razmaženiji hardeрaši mlje ove zemlje. Pojavile su se stoga i prve **off-line** vijesti koje kažu da nekadasnji **konkurenti** <4> sve više **žude za pripajanjem** <5> 'Mom mikru' i stoga pomno prate svaku moguću **promjenu u impresumu**" <6> (sva podvlačenja i cifre idu na dušu E. J.)

## Crno i masno

Šta se zapravo krije iza ovih poruka koje smo naglasili polucrnim slovima (grafičari ih nazivaju i crnim ili masnim), ili to već izaziva neke druge asocijacije)? Počnimo od druge poruke: „**Beogradska kompjuterska beletristika**" su, naravno, „Računari" i „Svet kompjutera", izdavači su BIGZ odnosno „Politika". „Beogradski", razume se, nije samo geografska odrednica nego i nešto drugo poznato, optužujuće. Daleko bilo! Zašto, uostalom, ne i „zagrebačke" — i to iz puno razloga pre ona nego beogradska — jer i tamo je bilo „kompjuterske beletristike", i to izveсno promašene („Male računale" i „Pilot video" više nisu na izlaze)? Mogao bi „Start", osim toga, da bude (samo)kritičniji i prema kompjutersko-video-hajfajskom glasilu svoje matične kuće („Vjesnik"), listu „Trend".

Prva poruka („**nisu uspjeli dostići**") obična je




insinuacija, a treća („**a jako su se trudili**") obična besmislica. „Računari" su uvek držali do svoje vlastite fizionomije i pre su mogli nastojati da se, i po koncepciji i po prezentaciji, razlikuju od „Mog mikra" nego da mu makar i malo nalikuju — kao što to odlično znaju čitaoci oba časopisa. Ne možemo a da na upitamo: za „Moj mikro" i njegov i bivši-sadašnji „novi čovjek" dosta o svom listu i sebi imaju tako nadmeno mišljenje da smatraju da ih svi moraju (makar i bezuspješno!) po svaku cenu kopirati? I zar nisu uspeli da nauče, da parafraziramo Bertranda Rasela, da je skromnost uvek više korisna nego štetna, a da je samozaljubljenost uvek više štetna nego korisna!

Najbolje dokaze za tu neverovatnu samozaljubljenost pružaju peta i šesta poruka. Ne samo da „Računari" „**žude**" da se pridruže božanskom „Mome mikru" nego oni čak vape da mu se „**pripoje**" da se u njega utope i izgube vlastiti identitet ako treba i život.

## Konkurencija i svrab

Tu ropsku pokornost „Računara" potkrepljuje bi šer misao o praćenju „**promjena u impresumu**" koja jugoslovenskoj računarskoj i drugoj javnosti poručuje da ovaj predstavnik „beogradske kompjuterske beletristike" ne samo da nije u stanju da ravnopravno razgovara recimo o nekom spajanju nego čak ni da nije kadar ni da zaslužı da bude obavešten o tome hoće li biti pripojen svom velikom bratu sa zapada, pa mu ostaje jedino da smerno pilji u impresum „Moga mikra" da bi saznao kakva mu je sudbina napisana: božanska milost, ili udarac nogom u poznato mesto. Jednom rečju kretenština!

Nikada nikome iz redakcije „Računara" nije palo na um, niti će se to ikada dogodili, da se ovaj časopis spaja sa „Mojim mikrom" ili mu se ne-daj bože pripaja! Ne bi nas, da budemo jasni do kraja, zanimala čak ni ideja da se „Moj mikro" pripaje „Računarima" — jer takve zamisli smatramo neozbiljnim. A saznajemo da ni „Svetu kompjutera" na ide loše. Ako smo time „ispucali", da ponovo upotrebimo formulaciju iz „Starta", „beogradsku kompjutersku beletristiku", za dalje detektivsko traganje za inicijatorima „pripajanja" ostaje nam jedino da sada parafraziramo i naše kolege: ljubljanska kompjuterska beletristika, tj. „Moj mikro". I bez obzira na „inicijatora" publikovanje takvih ideja i njihova eventualna razmena među bilo kojim „subjektima" može se jedino tretirati kao obična kombinatorika.

Motiv za čudne i ničim izazvane insinuacije na račun „Računara" najlakše se otkriva u formulacijama upotrebljenim u tekstu „Urednik i njegov mikro". Najindikativnije je ono bolešljivo insistiranje na terminu „**konkurenti**". U udžbenicima psihologije mogu se naći stručna objašnjenja za podsvesno pripisivanje vlastitih motiva drugima. Narod to, međutim, kaže mnogo jednostavnije, malo priprosto: svako se češe tamo gde ga svrbi.

*Esad Jakupović*

# Septembar u Minhenu

**Ako ste se odmorili od letovanja, pretrpeli šok od povećanja uvozne kvote za računar i ako vam je ostalo dovoljno para da priuštite sebi novog elektronskog mezimca, evo šta vas očekuje u gradu koji važi za „šangri la hakerskog šopinga".**





Pre svega, očekuje vas kiša. U Minhenu ne pada kiša samo onda kad pada sneg. Ova neprijatnost biće kompenzovana prijatnim bavarskim duhom koji za razliku od ostalih Nemaca ne doživljava svet (i kupovinu računara) tako ozbiljno i dosledno. Da bi vas odmah odobrovoljili uzeće vam samo 1% provizije za menjanje čekova ili valute.

Odmah kod glavne železničke stanice smeštene su radnje koje su cilj vašeg putovanja. Tu su znamenite ulice kao Beyerstrasse u kojoj možete kupiti kišobran za četiri marke (preko potrebno) ili digitalni telefon za sedam maraka (da se nađe). U blizini je i Schwanthalerstrasse gde u diskontnim prodavnicama, pored robe koja je ispala iz kamiona ili preko koje je prešao valjak, možete da nađete niz interesantnih stvari.

Ipak, pre nego što potrošite novac na sitnice za koje ćete kod kuće ustanoviti da vam nisu potrebne, najbolje je da se odmah uputite u središte zbivanja — znamenitu Schillerstrasse u kojoj su smeštene prodavnice svih trgovaca računarskom opremom koji drže do sebe. Radnje se, sa malim izuzecima, otvaraju u devet sati. Ako signale ranije stignete to se razgledanje, ako ste opremljeni nepropusnom (wasserdicht) obućom.

Veoma je korisno unapred znati šta želite da kupite, jer da vas odlaganje odluke se Minhen koštati mnogo živaca.

Gotovo sve prodavnice opremljene su simpatičnim zemljacima iz naših krajeva tako da vaš eventualni „hochdeutsch" neće baš previše doći do izražaja. Za uzvrat, možete na miru razgovarati s svemu što vas interesuje u vezi sa kupovinom računara. Uz obaveznu konstataciju da cene memorija i dalje divljaju, utešiće vas da su zato cene ostale računarske opreme pale, tako da vam opet izađe na isto kao i pre pola godine.

Ako ste želeli da kupite neki sistem koji se druži sa IBM-om (IBM udruživ), evo sa kakvim cenama možete da računate:

- obična osnovna XT ploča (bez memorije), 190—200 DEM,
- turbo XT ploča, 8 MHz, 220—250 DEM,
- turbo XT ploča, 10 MHz, 230—260 DEM,
- turbo XT ploča sa integrisanom multi I/O, 10 MHz, 300—350 DEM,
- AT osnovna ploča, 6/10 MHz 700 DEM (bez memorije),
- AT osnovna ploča 6/8/10/12 MHz, 800—900 DEM,
- kontroler disketnog pogona, 2x360 K, 60—70 DEM
- kontroler disketnog pogona, 2x360/720/1,2 M 110—140 DEM
- hard/flopi kontroler 2D+2H (WD), 300—380 DEM
- disketni pogon 360 K, 200—800 DEM,
- disketni pogon 1.2 M 280—320 DEM,
- XT kućište (AT-look) 100—190 DEM
- AT kućište 140—190 DEM,
- XT jedinica za napajanje 140—160 DEM,
- AT jedinica za napajanje 190—200 DEM
- hard disk, SEAGATE ili sličan (20M/65ms) 520—800 DEM
- hard disk, ST—251 ili sličan (40M/40 ms) 950—1200 DEM
- CGA grafička karta, 60—100 DEM
- Herkules grafička karta 100—120 DEM
- EGA i proširene EGA kartice, 300—650 DEM,
- multi video karta (CGA, HGC+printer) 200—250 DEM,
- multi I/O karta XT 120—190 DEM,
- I/O plus karta AT 110—190 DEM
- TTL monitor najniža cena oko 190—200 DEM,
- EGA monitor najniža cena oko 650 DEM,
- tastature širok raspon od 60—5000 DEM
- miš sa softverom 95—170 DEM
-

Nadamo se da smo ovim pokrili sve ono što bi moglo da vas interesuje. Ostale su još samo memorije čije su cene nepredvidive i menjaju se iz dana u dan. U vreme naše posete tipične cene za čip su bile 150 ns — 25 DEM, 120 ns — 33 DEM i 100 ns — 40 DEM. Za 256 kilobajta memorije vam je potrebno devet takvih čipova.

Cene kompletnih računara kreću se u rasponu od 1700—1800 DEM za XT i 3000—3200 za AT kompatibilni računar. U ove cene je uračunat kompletan sistem sa hard diskom, jednom disketnom jedinicom i herkules karticom i monitorom.

Želimo vam mnogo sreće pri kupovini i još više na carini. Usput carinici čitaju „Računare" i znaju sve o kompatibilcima — čak i razliku između 150 i 100 ns.

*Bata Data*





# Dobro

Ako sanjate da napravite robota i volite da se igrate sa lego setovima, upućujemo vas u hrabar potez firmi „Atari" i „Personal Robots" da naprave proizvod baziran na lego kockama.

## Ne daj se mala ruko

„Robokit Atari ST" sastoji se od štampane pločice koja se stavlja u računarski kertridž port, 360 kilobajta softvera i priručnika. Tome treba pridodati i „Basic Technical Lego set 1032" i „Educational Control Lego set 1090". To je sve. A košta oko 40 funti. Inače, „Personal Robots" firma koja je dizajnirala „Robokita", poznata je kompanija za industrijske robote. Samu prodaju preuzeo je „Atari". Šarmantni spoj lego imidža i „Robokit" imena, prvenstveno korišćenog u industrijskim sistemima, sam po sebi nameće zaključak da „atari ST" verzija koristi identični sistem programiranja i ima sve funkcije prethodnika iz produkcije firme „Personal Robots". Dizajniranje sopstvenog robota prikazano je u priručniku u pet verzija kroz ilustracije i dijagrame za konstrukciju. A ako ste trenutno neinspirisani šarenim lego kockicama, moguće je zaobići ih jer robokit interfejs uključuje čitav niz prekidača koji, naravno, pojma nemaju čime upravljaju.

„Robokit" interfejs ima solenoide, motore za kretanje i mikroprekidače. Na njega se mogu priključiti senzori za zvuk, svetlo, toplotu i dodir. Pa ako vaš stvaralački duh gine za metalnim robotom prave veličine, napravite ga. A možete koristiti i neke druge konstrukcione sisteme kao na primer one „Fischer-Technika" iako je, možda, lego najprimereniji jer omogućava rešenja vrlo slična veoma skupim sistemima. Konstrukcija „Robokit" robota odvija se na pet nivoa počev od građenja robota od lego seta, preko internog povezivanja, pa povezivanja sa interfejsom, konfigurisanja hardvera i softvera sve do same vožnje robota. No pre upuštanja u gradnju pravog modela neophodno je napraviti prekidače. Ovde je moguće pustiti mašti na volju i maksimalno iskoristiti vlastitu strast za „budženjem" praveći ih od ispegla na spajalice, tanka folije, elastične trake ili nezaobilaznih lego kockica.

Da konstruisanje robota nije šala, osobito lego sistema za upravljanje, dovoljno sklonim havarijama jasno je i najvećim optimistima kada robot katastrofično počne da jurca i skonča transformisan u gomilu rasutih kockica. Prema iskustvu Nika Volkera projekat „mala ruke" uz najbolju volju i trudoljubivost uzme oko desel sati samo da se podigne i krene. Već sa sledećim je lakše, pojede samo oko šest sati.

## Pet modela

Priručnik iz „Robokit Atari ST" kompleta predstavlja pet modela:

1) **Malu ruku**, koja koristi samo 2 motora radeći isti posao kao i tromotorni dizajn. Jedan motor je zadužen da je vodi oko njene baze, odnosno fiksne tačke, dok je drugi podiže i spušta, istovremeno otvarajući i upravljajući hvataćem. Baza i sama ruka, odnosno zglob i rame, imaju ugrađene prekidačke senzore. Ovo „maloj ruci" omogućava da bude vožena od strane programa, kao i da odgovara na spoljne faktore.

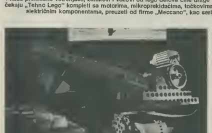
2) **LIFT-operater** koji uči principe različite povratne sprege, približava se i daje priliku eksperimentisanju sa algoritmima za podizanje, što je od značaja osobito za one koji prigovaraju da su dotični glupavi.

3) **Card reader**, čitač kartice koji koristi staru, dobru tehniku bušenih kartica.

4) **Jednostavni ploter** koji koristi dva lego motore umotana u lepljivu plastičnu traku kao sa vaših patika tako da formira valjak i elektromagnet sopstvene izrade od 800 namotaja žice oko 25 milimetara duge plastične cevi. Radi na dva motora: jedan ima zadatak da vozi valjak, a drugi da pomera pero

# ...došli u svet robotike

**Da su lego kockice najsjajnija igračka za konstruisanje poznato je svim dečacima i devojčicama koji su bar jednom probali da ideje iz mašte i zadatih šema izgrade lego sistemima. Za razliku od drugih igračaka za konstruisanje, lego raste zajedno sa svojim vlasnicima i ne gubi na aktuelnosti. Tako danas, kada ste porasli i kada prekrasne vetrenjače, zamkovi i vozovi od lego delova čine drage uspomene ranog detinjstva, na vas čekaju „Tehno Lego" kompleti sa motorima, mikroprekidačima, točkovima, senzorima i drugim, najrazličitijim električnim komponentama, preuzeti od firme „Meccano", kao serije sistema za konstruisanje.**



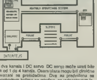
Elektromagnet je tu da podigne i spusti narečeno pero.

**5) Maksi ruka** je najambiciozniji robokit projekat. Kromotoma ruka koja sem da se kružno pomere, može i da se savije i ispruži u dužinu od ramena. Sposobna je da hvata i premešta objekte bilo gde u velikom prostoru. Toliko je svestrana da može da pokreće figure u šahu što je jako lepo ako volite da igrate. A volite, zar ne?

## Povezivanje sa interfejsom

Iako je problem internog povezivanja u teoriji jedan od najjednostavnijih nivoa u konstrukciji robota, javljaju se dva problema koji ih čine komplikovanijim nego što je to opisano u priručniku. Pre svega, mnoge veze su tako duboko zakopane u model da ne postoji zdrava mogućnost da se do njih dođe. Naravno, ako ne nameravate da strateški provalite u model, napravite vezu i uz pomoć trikova sve vratite u prvobitno stanje. Drugi problem je blizak daltonisbme i iskazan je u lego žicama jedne boje, pa veze na motoru nisu označene istorijskim „plus i minus". Kada se zna da i najprostiji robot koji ćete graditi ima minimalno dvanaest žica za povezivanje, ovo postaje tužan problem. Od sveg srca preporučujemo sjajno rešenje Nika Volkera, koji je lepo zalamio lego konektore za komedic višebojnog trakastog kabla i vredno zapisao šta koja žica znači.

Povezivanje sa interfejsom je jednostavno. Ima 16 kanala, 8 za ulaz, 8 za izlaz. Područje za prototipsko povezivanje je, takođe, obezbeđeno za povezivanje dodatnih kola. Od 8 izlaza svaki može biti korišćen kao logički ili pak kao izlaz za napajanje. No bitno je imati na umu da dvosmerni motori traže-




ju dva kanala i DC servo. DC servo može uzeti bilo šta od 1 do 4 kanala. Osam ulaza mogu biti direktno povezani sa prekidačima. Dva su predviđena sa neophodnim kolima za interfejs, sa jednostavnim lego optičkim senzorima. Broj opcija može biti dodat štampanoj ploči. Ovo uključuje i dodatne jedinice za napajanje većih motora za hladnjake i pobudne čipove spoljašnjih motora. To je sve. A sada možemo preći na problem konfiguracije. Jer, model je konstruisan i povezan, ostaje još samo da se pokrene ST-om. Nakon butovanja diska javlje se pitanja izbora „skrin back dropa", odnosno konfiguracija ekrana iz menija od *.RSU fajlova. Oni su baš kao i GEM-ov desk top .inf fajl koji sadrža detalje rasporeda na ekranu kao i raspoloživih slika. Izborom pomoću miša na „cancel" vodite kroz čist GEM ekran. U gornjem levom uglu je rupa za miša. Ovim izborom proizvodi se slika uspavanog miša i uključuju se sva ekranska prikazivanja stanja. U kompleksnoj kontrolnoj situaciji to omogućava ST-u da celu snagu procesora posveti kontrolisanju i komuniciranju sa robokit interfejsom. „Backdrop" meni iz ovog ekrana uključuje opciju da se napuni novi grafički background. Mogući formati fajla za ovaj Backdrop su GEMPaint, Degas ili NeoChrome. Punjenje background slike jeste integralni deo robokit softvera.

## Kolekcija za vožnju

Biranjem „Add Icon" opcije iz „Icon" menija pojavljuje se ekran pun definisanih ikona i dugmića za pritiskanje. Tu su i gornja i donje strelice, start, stop ali i više generalisane ikone kao ljudi, kola i signali napajanja. Izabirajući jedan, može se postaviti bilo gde na pozadinu. Posle dobijanja željene slike i pridruživanja značenja, ostaje da se izborom ikona robot stavi pod kontrolu. Što bi rekli, klikneš na ekranu, a robot radi ono što si kliknuo. Meniji omogućavaju start sistemskog sata, pa onda svi delovi sistema primaju sinhronizovane vremenske impulse. Važno je i inicijalizovati motore tako da robokit zna koji su motori povezani sa kojim izlaznim linijama. Ostaje nam programiranje. Može se slobodno reći da se pisanje programa za vožnje robota radi na sličan način kao što se pišu programi za „Omnis Quartz Data Base Manager": sve naredbe se biraju iz menija da bi se osigurala sintaksna korektnost svih programa, kao i da se smanji količina rada robokit interpretera. Poruke su podeljene u logičke grupe, što pojednostavljuje nalaženje vlastitog puta kroz razna menija. Imena poruka su slična ikonama iz C programa kao što su „move-absolute", „move-time", „init-motor". Posle izabiranja poruke iz menija ona se pojavi u edit prostoru tako da bilo koja varijabla može biti promenljiva. Poruka se tada šalje direktno robotu ili području za prikaz poruka znanog kao sakvenos.

Okruženje za programiranje koje je korišćeno za kontrolisanje „Personal Robots" produkta nazvano je PROSE (Personal Robots Operating System & Environment). PROSE, inače, obuhvata mnogo više od same kolekcije za vožnju robota. Ima prilično sveobuhvatan programski jezik opšte namene, operativni sistem dizajniran za kontrolu u realnom vremenu i GEM interfejs. „Personal Robots" je učinio značajan napor da iskoristi sva funkcija GEM-a, tako da se skoro svim funkcijama GEM-a može prići iz PROSE. Skala sekvenci u PROSE pripada korisnički definisanoj jedinici nazvanoj „slot". Svaki „slot" ima mogućnost za jedan određeni aspekt kontrole robota. Pre dizajniranja jednog PROSE programa bitno je odrediti koji aspekti robota zahtevaju istovremeni nadzor. Tako, na primer, mikro miš kao robot trebaće jedan slot za vožnje točkova, drugi slot za upravljanje i treći za nadziranje sudara. Raspoređivač slotova omogućava svim slotovima da se pojava kao da rade jednovremeno. Slotovi takođe mogu slati poruke jedan drugome, tako da, recimo, kada slot za sudare uoči prepreku, on informiše slot za točkova da ih zaustavi. Divno, zar ne? Osobito ako ste spremni da date 79,95 funti za robokita i još 40—80 funti za lego kockice (telefon „Atanja" je 0753 33344). Kao alat za podešavanje, sa profinjenim softverskim okruženjem, robokit predstavlja sjajan uvod u svet robotike. Plus što omogućava da i nadalje negujete svoju ljubav prema lego kockicama bez opasnosti da vas optuže da ste načisto podetinjili.

*Vesna Ćosić*

Za „Računare" govori Čedomir Mraz

# Milion osmeha Čede Mraza

**Znate li onu priču? Kao, usplahirana majka obigrava oko svoga dragog deteta i crveneći, sa puno nelagodnosti kaže: Znaš dušo, ti si sada porasla i ja, ovaj, zbilja moram da te uputim, uh, da ti kažem *(Jao)*, baš mi je nezgodno! možda je to ipak otac trebao da ti saopšti; mislim *(koluta očima)*, ti si na pragu puberteta i ovaj *(CCC)*, kako da ti kažem, stvar je vrlo delikatna; *(znoji se na mukama)* možda sam te ipak trebala pripremiti!? *(UZDAH)* Ti si velika, znaš . . . *(kulminacija porodična dramatika, ubrzano, gubi dah)* slušaj me — *Deda Mraz ne postoji!!!! (klona iscrpljeno)* — Vrlo važno! Ako ne postoji Deda Mraz, postoji Čeda Mraz! — hladno odgovori dete i lakim hakerskim hodom parkira se za računar iz radnje „Mraz Elektronik". — A Deda Mraz i onako nije imao veze sa kompjuterima — dodade.**

***Računari:** Ko je u stvari Čeda Mraz?*

**Č. Mraz:** Čeda Mraz?! Sebe smatram potpuno prosečnim čovekom sa velikim interesovanjem za tehniku; znači tehnika je od malena bila žarište mog interesovanja i kao dete od 5—6 godina sam već znao, ili sam mislio da znam da ću postati inženjer i baviti se tehnikom.

***Računari:** Bili ste beogradski đak.*

**Č. Mraz:** Da. Završio sam fakultet, odnosno prvo tehničku školu „Nikola Tesla" i onda fakultet ovde u Beogradu.

***Računari:** Kakva je onda bila škola „Nikola Tesla"?*

**Č. Mraz:** Pa ja sam bio jako zadovoljan, jer smo učili bilo je puno stručnih predmeta i učilo se ono što jedan tehničar treba da zna. I mislim da mi je to dalo jednu vrlo solidnu bazu za dalju nadgradnju. Posle sam se zainteresovao za kompjutersku tehniku ... Pošto je to kod nas još bilo u povoju, tamo sedamdesetih godina, odlučio sam da odem u Zapadnu Nemačku. Da bih to specijalizovao, zaposlio se kod firme „Simens" i od sedamdeset druge neprestano do osamdeset druge bio kod „Simensa" kao inženjer održavanja procesnih računara. To su kompjuteri koji upravljaju procesom proizvodnje fabrika, električnih centrala. Ja sam sa grupom od pet ljudi preuzeo odgovornost za saobraćajne kompjutere u Minhenu kojih je tada bilo sedam povezanih u mreži.

***Računari:** Sada ste biznismen. Kakav je bio početak?*

**Č. Mraz.** Početak je, razume se, bio kod „Simensa", što se tiče stručnog usavršavanja u toj kompjuterskoj tehnici. A posle više godina, skoro deset godina boravka kod „Simensa" bilo je jasno da neku veliku perspektivu u jednoj takvoj firmi dalje nemam, jer je takva firma zainteresovana da, razume se, iskoristi one kvalitete zaposlenih ljudi koji firmi najviše odgovaraju. Znači da konkretno radim na određenim tehničkim problemima itd. A ja sam tu hteo već malo više! I tada sam osnovao sopstvenu firmu „Mraz Elektronik" gde je dolazilo do izražaja ne samo jedna uska usmerenost na kompjutersku tehniku nego celokupno relativno široko tehničko obrazovanje koje sam stekao tokom školovanja.

***Računari:** Kako izlazite na kraj sa konkurencijom? Ovde se priča da vam izvesni Zemler (Seemuler) i gospodin Herman blago zagorčavaju život?*

## Sve za mušteriju

**Č. Mraz:** Pa, ne mogu reći da mi bilo ko zagorčava život, jer sa konkurencijom nema problema. U kom smislu? Kod mene dolaze ljudi koji su obavešteni uglavnom preko preporuka ili preko reklama i koji traže na našem jeziku objašnjenje, demonstraciju i eventualno malo i obuku, tako da se ne bojim toga da ću ostati bez posla. Imam više problema da savladam sav obim posla koji dolazi nego da se bavim nekim drugim problemima! Firma sada ima pored mene još troje zaposlenih i nekoliko spoljnih saradnika. Obzirom na potražnju i potrebe, razmišljam da je proširim, što međutim dovodi i do jedne druge problematike: naći odgovarajuće kadrove koji bi se istim žarom radili korektno i poslovno, jer moja filozofija je da ljudi treba da budu zadovoljni. Za celokupno vreme mog poslovanja bilo je, razume se, reklamacija, ali smo sve reklamacije i sve probleme rešili. Nije mi poznat ni jedan slučaj da je neko na kraju bio nezadovoljan. Svaku mušteriju pokušavamo da zadovoljimo, jer nam je to i najbolja reklama. Kroz dobre kontakte sa stručnjacima iz cele Jugoslavije uspeo sam da organizujem mrežu servisa u celoj Jugoslaviji pa se ovako, zajedničkim snagama trudimo da sve probleme rešimo na adekvatan način.



Biznis sa Jugoslovenima: Čedomir Mraz

**• Događa se da dođe cela porodica, roditelji i deta i baka i deda, i kupe celu konfiguraciju. Jer samo na taj način mogu legalno da uvezu kompjuter u našu zemlju.**

***Računari:** Prema informacijama ovdašnje kompjuterske underground scene vaše najbrojnije mušterije su Jugosloveni.*

**Č. Mraz:** Dakle, kod mene dolaze stvarno različiti kupci! Počevši od roditelja koji dođu sa decom da bi kupili personalni kompjuter da dete počne da radi, a mislim da je veliki broj roditelja svestan neophodnosti ulaska dece u najnoviju tehnologiju, pa do pravoslavnih sveštenika.

***Računari:** Moram priznati da sam veoma radoznala koja računare koriste sveštena lica?*

**Č. Mraz.** PC, naravno. Obzirom da je cena osnovne konfiguracije personalnih kompjutera oko hiljadu maraka, dostupni su već kao početni kompjuter. To je nekada bio „sinkler" i „komodor", a danas je to praktično postao personalni kompjuter PC kompatibilan XT. Međutim, dolaze i profesionalci koji se snabdevaju i boljim kompjuterima, dolaze servisi koji se snabdevaju rezervnim delovima, kompjuterskim komponentama, ljudi iz male privrede, a imam poslovne kontakte i sa različitim institutima, fakultetima itd. koji takođe nabavljaju opremu za svoje računske centre.

***Računari:** Znači, dolaze vam i ljudi koje možemo svrstati u kategoriju korisnika, mislim da nisu profesionalni softveraši, hardveraši, već lekari, biolozi, prevodioci, muzičari, sveštenici.*

## Jugoslavija u radnji

**Č. Mraz:** Da. Evo na primer jednom su došli sveštenici sa vrlo jasnim zahtevom: mislim PC štampač ... Nije prošlo mnogo vremena a pojavile su se njihove kolege pa kasnije druge i tako stvar se razčula. Tako je i sa drugim profesijama. Sve je više i više prevodioca, pisaca, lekara. Javljaju se, traže savet i ja gledam da za svakog korisnika odaberem onakvu konfiguraciju koja mu najviše odgovara. Znači, u takvim slučajevima mora se voditi računa u prvom redu o finansijskoj mogućnosti a u drugom redu šta je za datu profesiju potrebno. Ako je, recimo, u pitanju prevodilac, razume se da mora da ima jedan kvalitetniji štampač jer će mnogo da piše, dvadesetčetvoro-pinski itd., monitor koji neće da zamara oči, dok za neke druge slučajeve ima drugih vrsta monitora, printera, pa čak i kompjutera. Jedna brzina ... veća ili manja, to u zavisnosti od namene i primene.

***Računari:** Priča se da ste svojim zemljacima prodali između 3000 i 5000 računara. Najekstremniji čak tvrde da se ova impozantna brojka odnosi samo na kompjutere u Beogradu i njegovoj okolini, recimo do Novog Sada.*

**Č. Mraz.** Ne bih rekao da je toliki broj, ali oko tri hiljade, to može da bude tačno, međutim, ne samo u Beogradu nego širom cele Jugoslavije. Ja sam veoma srećan da kod mene dolaze naši zemljaci iz cele zemlje. Iz Skoplja, Novog Sada, Beograda, Sarajeva, Zagreba, Slovenije, Splita, Rijeke, iz svih krajeva zemlje dolaze kod mene i nemam nikakvih problema.

***Računari:** Onda ćete vi da nas uједinite kako stvari stoje i kompjuteri ... Naširoko je rasprostranjeno*



Građanin sveta: Čeda Mraz sa suprugom

*jedno laskavo mnenje, da je Čeda Mraz prodao više računara u Jugoslaviji nego svi jugoslovenski proizvođači zajedno?!*

**Č. Mraz:** O tome ne mogu ništa da kažem, jer ne znam koliko jugoslovenski proizvođači proizvode. Ja sam veoma srećan kad bilo kome mogu da pomognem i smatram da svaki računar koji se kupi i uveze u Jugoslaviju čini našu zemlju bogatijom, ne materijalno, pri tome ne mislim da materijalno odbacujem, nego mnogo više cenim vrednost koju ima kompjuter za unapređenje opšte tehničke kulture i dalji prosperitet naše zemlje.

***Računari:*** *Bilo kako bilo, vi ste čovek koji je uveo kompjutere u hiljadu kuća i kao takvi direktno zaslužan za podizanje nivoa domaće hardversko-softverske svesti. Jer ljudi na njima rade, stvaraju i šire informatički nivo, smanjujući opštu razvojno tehnološku zaostalost koja je, svidelo nam se ili ne, naša stvarnost. Kako se osećate?*

**Č. Mraz:** Pa srećan sam! Često dolaze mladi ljudi da pokažu i pričaju o programima i stvarima koje su uradili sami. To čini veliku satisfakciju, videti koliko su naši mladi ljudi sposobni i kako relativno malo treba da bi se ta naša tehnološka zaostalost vrlo brzo nadoknadila.

## Centar za informacije

***Računari:*** *Neki misle drugačije. Javno se tvrdi da vi destabilišete jugoslovensku računarsku industriju jer vaši kupci unose računare u Jugoslaviju bez ikakve kontrole!*

**Č. Mraz:** Pa, sa tom konstatacijom se ni u kom slučaju ne slažem! Kad bi naša industrija bila u mogućnosti da pruži potencijalnom kupcu kompjuter u okviru od hiljadu do hiljadu dvesta maraka, kompletnu konfiguraciju sa monitorom, verujem da niko ne bi putovao u Nemačku i trošio još par stotina maraka za put i nekoliko dana gubljenja vremena da bi uvezao kompjuter. Pored toga, smatram da naša zemlja ne samo da ima kočnicu o kojoj smo već govorili, tehnološkog napretka, već ima i relativno veliku carinsku stopu koja ide preko 50% na vrednost kompjutera, i ljudi se dovijaju. Moraju kompjutere kupovati u delovima, odnosno posebno hard disk na jednu osobu, kompjuter na drugu osobu, a ako je neki bolji monitor, EGA, na treću osobu, printer na četvrtu. Čudo jedno! Dogodi se da se dođe cela porodica: roditelji i deca i baka i deda i kupe celu konfiguraciju. Jer samo na taj način mogu legalno da uvezu kompjuter u našu zemlju.



**• Lepa žena, ili prijatan osmeh, otvaraju... otvaraju svačija vrata. I ako je potrebna pomoć, razume se da se je ne oglušim.**

***Računari.*** *To je zbilja komplikovano. A ne postoji ni mesto, centar gde običan građanin kupac, može ne samo da kupi, nego gde može da se informiše, da proba, da nauči, da pita. I to preko cele godine.*

**Č. Mraz.** Mogu danas obelodaniti da sam napravio jedan dogovor sa Domom omladine u Beogradu o informacionom centru za omladinu i potencijalne kupce, baš zbog toga što ste vrlo tačno konstatovali da sada takve institucije ne postoje. Moram da priznam da sam bio veoma uzbuđen i radostan kada mi se Dom omladine obratio sa željom da se jedan takav centar osnuje u centru Beograda. I mi smo u tom smislu napravili dogovor i uskoro će taj centar početi da radi.

***Računari:*** *Kada, konkretno?*

**Č. Mraz:** Ja mislim početkom oktobra.

***Računari:*** *Kako se sad kreću cene za XT i AT mašine?*

**Č. Mraz:** Cene za XT — početna konfiguracija je ispod hiljadu maraka, AT je ispod dve hiljade maraka. U zadnje vreme se cene nisu bitnije menjale zato što je cena memorijskim čipovima nešto porasla, dok je cena hardverskim pločicama nešto pala, tako da se to kompenzovalo i cene su relativno stabilne već jedno šest sedam meseci. Sad nastaje nov problem, jer je dolar porastao zadnjih par meseci za oko deset posto u odnosu na marku, tako da je pitanje da li će kompjuteri ostati u Nemačkoj sa istom cenom kao što je do sada bilo ili će biti drugačije. Mogu da napomenem da su japanski štampači takođe poskupeli od 5 do 10, čak i dvadeset posto. I to zato što je Evropska zajednica zbog damping cena japanskih štampača podigla carinu na njihov uvoz za 5—30 posto.

## Cene u porastu

***Računari:*** *Je li istina da se XT sve manje traži, a da je to i dalje perspektiva sada vrlo traženog AT?*

**Č. Mraz:** XT se još uvek traži, ne toliko kao pre, a AT se u principu sve više i više traži. Ja mislim da je sada otprilike pola-pola.

***Računari:*** *Iako ste, u principu jeftiniji od mnogih isporučilaca, cene memorija su kod vas nešto više kao i cene pojedinih monitora. Zašto?*

**Č. Mraz.** Poznajući problematiku našeg tržišta i našeg kupca, primorani smo da koristimo isključivo kvalitetne elemente. Pločice u svim kompjuterima su približno istog kvaliteta. Proizvode se u Tajvanu, Hong-Kongu, Singapuru. Postoje memorije različitog kvaliteta. U mojoj radnji se prodaju samo kvalitetne, jer znamo da ako nastupi greška u kompjuteru i kupac je time nezadovoljan, traži usluge naših servisa, nervira se, to nije pravo rešenje. Ista stvar je i sa monitorima. Postoji monitori koji su lošijeg kvaliteta. Ja uglavnom prodajem „Philips", „Visa" i još neke boljeg kvaliteta. Lošije monitore ne prodajemo. Kupci koji žele jedan slabiji monitor mogu da ga kupe kod konkurencije, što se ponekad i dešava.

***Računari.*** *Koja je vaša procena primene OS/2 i šta mislite kakva mu je perspektiva u Jugoslaviji a i šire?*

**Č. Mraz:** Za sada je interesovanje minimalno, ali je sigurno da će u dogledno vreme biti sve veće i veće. Perspektiva je sigurno tu! Čim je IBM objavio taj sistem, perspektiva mu je zagarantovana.

***Računari:*** *Šta je sa 386?*

**Č. Mraz.** Na žalost, ta je mašina još uvek veoma skupa i za naše ljude prilično nedostupna. Što ne znači da to ne kupuju instituti za profesionalnu primenu.

***Računari:*** *Kako stoje stvari sa memorijama? Šta mislite kako će se kretati cene?*

**Č. Mraz.** Cene momentalno rastu! Još uvek vlada tendencija porasta, znači još nije na vidiku pad cena! Kad pomislimo da isti čipovi koji su pre godinu dana koštali 4—5—6 maraka danas koštaju 18—20 maraka, jasno je koliki je to problem. Pojavom memorije od 1 Mbita se situacija nešto promenila, međutim, 1 Mbitni čipovi mogu se koristiti isključivo u AT u gde postoji još jedna nezgoda: takav AT mora da ima bar 2 Mbyte-a memorije na osnovnoj ploči. Znači sa jednim megabajtom sa tim 1 Mbit čipovima to ne funkcioniše.

***Računari:*** *Moram da vam ispričam jednu anegdotu. Jedan moj prijatelj, koji je kod vas kupio AT (pre toga vam je bio mušterija za XT) jezivo je ljut i žali se na sva usta da su mu memorije spore, da mora da isključi turbo, da uopšte nema računar sa datim performansama, jer mu radi na 8, a neće na 12 MHz. A jedan drugi je više nego zadovoljan. Kaže ima bolje memorije nego što je predviđeno konfiguracijom. Naime, radi mu na 16 MHz.*

**• Najvažnije je da mi moj posao čini zadovoljstvo, da mi je dobro, da nemam problema, niti sa mušterijama, niti sa mojim prijateljima, niti sa bilo kojim institucijama...**

## Provere i garancija

**Č. Mraz:** AT kompjuteri mogu da rade od 8 do momentalno 16 MHz. Zavisno od toga kojom brzinom želi da se radi moraju se kupiti odgovarajuće memorije. Memorije od 60 ns potrebne za ovaj računar od 16 MHz koštaju trenutno u Nemačkoj preko 30 maraka. Prema tome, vaš prijatelj koji je nezadovoljan radom svog računara može kod mene bez problema da svoje memorije zameni za brže. Do sada se nije dogodilo da je došla bilo koja mušterija koja je kod mene bilo šta kupila, da li je to Herkules ili kolor karta ili multifunkcijska, i da je tražila zamenu uz obrazloženje i da to nije ušlišeno.

***Računari:*** *Šta se to zbiva na tržištu memorija? Jer ovo se ne dešava samo vama. Čula sam da ste svojevremeno kupili veću količinu memorija od renomiranog proizvođača i da, iako sve lepo piše i imaju dobre karakteristike, na primer 100 ns, neće da rade u AT ni na 6 MHz.*

**Č. Mraz:** Problem memorije nije tako jednostavan. To su dinamička memorije koje imaju refresh ciklus, znači ciklus obnavljanja stanja. Taj ciklus je kod AT a definisan na jedan način i ako neka memorije ne



Čevek u modi: Mraz sa kćerkom

rade u AT-u to ne znači da u drugim sistemima sa nekim drugim ciklusom refresh a neće raditi. Ne tom problemu radimo da li ćemo ga rešili pitanje je

*Računari: To se verovatno menja od jedne do druge proizvodne serije*

**Č. Mraz** Baš tako. Javljaju se memorije koje nisu loše ali u pločama AT-a ne funkcionišu

*Računari: Da li vi atestirate svaki računar koji prodajete? Ovo me zanima jer to zahteva ogromno vreme, nekoliko sati po mašini pa kad se pomnoži sa brojem koje plasirate onda se plasman vrlo usporava.*

**Č. Mraz:** Ja kompjuter testiram pre nego što ga demonstriram mušteriji zatim ga demonstriram i testiram kada dođe mušterija i pored toga, kao treće, dajem garanciju u Jugoslaviji. Takva poslovna filozofija se pokazala ispravnom jer prilikom prvog testa i demonstracije dolazi do otkrivanja skoro svih mogućih grešaka tako da servisi imaju relativno malo probleme

*Računari. Bije vas dobar glas da imate najrazvijeniju i najekspeditivniju servisnu službu u zemlji. U maju ste održali u Opatiji seminar za svoje servisere. Kako ih odabirate?*

**Č. Mraz** Ne samo svoje servisere, nego i svoje mušterije smatram za prijatelje i tako se i ophodim prema njima. Možda je prednost moje firme u tome što je na trećem spratu ne dolazi preveliki broj kupaca, tako da mi uspemo da posvetimo dosta vremena našim kupcima. Serviseri koji su kao što sam rekao, moji prijatelji veoma rado pomažu svakoj mušteriji koja ima probleme sa svojim računarom. Seminar koji smo održali u Opatiji koji je bio razume se i didaktičan uz određene demonstracije predavanje i tako dalje, imao je i cilj koji je po meni čak i mnogo važniji pošto ti serviseri i tako znaju svoj posao a taj cilj je bio da nas zbliži. Da se svi upoznaju

## Spekulacije o zaradi

*Računari: Smatra se da ste biznismen od zanata. Kakve to implikacije ima u nemačkoj sredini imajući u vidu da vi tamo delujete i kao reeksport odnosno solidan izvoznik pre svega potpomažući na taj način devizni bilans SR Nemačke. Mislim pre svega na beneficije*

**Č. Mraz.** U Nemačkoj je takav sistem da firme koje se bave poslovanjem nemaju nikakve beneficije bilo da importuju ili da eksportuju — što se tiče države. Što se tiče banke obzirom da uvek ispunjavam na vreme svoje obaveze. Imam odlične poslovne veze sa našim bankama sa kojima poslujem a što se tiče kontakata sa drugim firmama oni su potpuno zadovoljavajući bez obzira što su to konkurentske firme u pitanju. Jer se u Nemačkoj, u principu, ceni svačiji rad i trud bez obzira da li je u pitanju Jugosloven ili Nemac, jugoslovenska firma ili nemačka. To je moje iskustvo

*Računari: Da ste među domaćim hakerskim establišmentom nepatvorena medijska zvezda vidi se i po senim nagađanjima oko vaše zarade. Šuške se da ste samo na Jugoslovenima zaradili između milion i tri miliona maraka. Nije li to malo preterano?*

**Č. Mraz:** Pa ja bih bio srećan da je to istina. ***(Slatko se nasmejao, a i ja sa njim).*** Ali, na žalost, to nije istina.

*Računari: Stvarno?*

**Č. Mraz.** Pa moram da kažem da mi taj finansijski efekat nije najbitniji u celom poslovanju. U prvom redu ja nisam trgovac već sam inženjer i tako i poslujem. Najvažnije mi je da mi moj posao čini zadovoljstvo da mi je dobro da nemam probleme niti sa mušterijama niti sa mojim prijateljima niti sa bilo kojim institucijama. Ja znam da to da li ću da zaradim par hiljada maraka mesečno manje ili više neće da me usreći ako ne mogu da budem srećan sa onim koliko mi je dovoljno da vrlo pristojno živim. Neko veliko bogatstvo me sigurno neće usrećiti i ne težim tome!

*Računari. Dok je na Zapadu statusna stvar težiti i negovati bogatstvo u ovim našim krajevima te stvari ne prolaze baš najbolje. Otprilike ako tamo nekom loše ide svi od njega beže a ako mu baš ide ima podršku iz sve snage. Doduše i ovde se stvari pomeraju u smislu tretmana onih koji imaju i umeju. U svakom slučaju vi ste bogat čovek. To mora da je jako ugodno*

> ▪ **XT se još uvek traži, na toliko kao pre, a AT se u principu sve više i više traži. Ja mislim da je sada otprilike pola-pola.**

## „Asketa" u „mercedesu"

**Č. Mraz:** U prvom redu smatram da iza mog dobro stojećeg i finansijskog stanja, stoji veoma puno rada. Smatrao bih se bogatim čovekom kada bih živeo a da uopšte ne radim. Međutim to nije moj cilj

*Računari. Verovatno ne biste ni mogli da ne radite, mislim ne ličite mi baš na lenjivca*

**Č. Mraz** Tako je. ***(Veselo se nasmejao)*** I nisam verujte

*Računari. Razmišljate li možda o zajedničkim ulaganjima u Jugoslaviji, odnosno o perspektivi da se deo novca koji ste zaradili okrene na obostranu korist ovde kroz nekakav projekat softverske, odnosno hardverske kuće. Jer kako stvari stoje to je nekakav ovdašnji trend menjaju se zakoni a i svest o perspektivi takvih zajedničkih ulaganja. Evo, najnoviji primer su CIP i Kanađani koji uveliko rade na sprovođenju jedne takve poslovne zamisli*

**Č. Mraz** Uvek sam otvoren za svaki razgovor i za svaki posao koji može da bude interesantan i koristan. Naravno ukoliko naši zakoni i propisi dozvoljavaju učešće sa mogućnošću kontrole sa moje strane tog poslovanja. To je bitno jer na taj način mogu da garantujem da će biti obezbeđeno jedno korektno poslovanje. Mislim da drugih prepreka nema. I mogu vam reći da mi se ta ideja čini veoma zanimljivom!

*Računari. Jugoslavija je s jedne strane prema stručnosti softveraša i hardveraša prava sila a sa druge strane radni sat jednog računarskog asa, gledano u svetskim okvirima, užasno je jeftin. Negde sam čitala da se radni sat programera u SAD vrednuje između 25 i 600 dolara. Javna je tajna da CIP i Kanađani svoju softversku kuću baziraju i na primamljivoj ponudi stručnjacima od hiljadu dolara mesečno preračunato u dinare što je u okviru svetskog tržišta cena radne snage više nego povoljno jer je radni sat domaćeg stručnjaka vrednovan veoma skromno. Kako stoje stvari u Nemačkoj?*

**Č. Mraz** U svakom slučaju su u Nemačkoj softveraši bolje plaćeni nego hardveraši. To što su naše plate softveraša relativno niske odraz je stanja opšteg nivoa plata u Jugoslaviji koji je nizak.

*Računari: Čeda Mraz je građanin sveta danas u Beogradu sutra u Singapuru. A ne mrzi vas ni da odjurite do Amerike pre nego što dođete kući u Nemačku ako to posao zahteva. Gde se najbolje osećate?*

**Č. Mraz:** U Jugoslaviji

*Računari. Zašto?*

**Č. Mraz** U prvom redu zato što sam Jugosloven u drugom redu zato što imam jako puno prijatelja sa kojima najlepše vreme mogu da provedem uz priču i dobro jelo baš kod nas u Jugoslaviji

*Računari. Jedan hakerski as tvrdi da niste baš asketa. Navodno vozite besnog „mercedesa" i to vam lepo stoji.*

## Popust za lepe žene

**Č. Mraz:** E, tačno. Posle dužeg odnosno višemesečnog razmišljanja traženja gledanja i tako dalje ipak sam se odlučio da kupim jedan „mercedes" stvarno lep i izuzetno komforan. A relativno i puno putujem kolima najčešće poslovno. To je na zapadu investicija koje se odbija od poreza i samim tim meni potpuno dostupna

*Računari. Da ste pravi star među domaćim kompjuterašima vidi se i po tome što naspram Čeda Mraza niko nije ravnodušan. Ili vas vole ili mrze. Svi vas znaju. Nije li razvoju vašeg biznisa jako pogodovalo i to što mušterije mogu slatko da se ispričaju i povere vam u svako doba na svom jeziku šta to zapravo hoće i koji im je predmet želja?*

**Č. Mraz** U svakom slučaju i to je osnova moje poslovne filozofije da upravo kupcu pomognem na našem jeziku adekvatno njegovim potrebama. A nije ni to uvek baš jednostavno. Eto jednom mi je došao jedan čovek iz Prokuplja kaže hoće jedan sasvim jednostavan računar za dete al da sve bude odmah sutra spremno. Ja se potrudim pripremim jednu jednostavnu konfiguraciju i on dođe. Kad pogleda on pa kaže. Jao pa ovaj monitor je mnogo lepši pa ova tastatura je mnogo bolja, jeste slažem se ja ali podsetim ga vi ste tražili nešto jednostavno da nije skupo. Šta da vam pričam insistirao je čak i kućište da promenimo

*Računari. Kada sam pripremila ovaj intervju čula sam jednu sasvim šašavu stvar. Naime, kao, Čeda Mraz daje na sve komponente specijalni popust lepim ženama!?*

**Č. Mraz** ***(Smeh)*** Lepa žena, ili prijatan osmeh, otvaraju otvaraju svačija vrata. I ako je potrebna pomoć razume se da se ja ne oglušim

*Računari. Ogovaraju vas da ste plejboj. Imate li šta da kažete u svoju odbranu?*

**Č. Mraz:** E to nije tačno! Jer kao što sam već spomenuo bogatim čovekom i plejbojem bih se možda osećao kada ne bih radio nego samo uživao što na žalost nije slučaj ili, na sreću

*Računari. Bez obzira, vi ste čovek u modi. Koliko će to još da traje?*

**Č. Mraz** Trajaće toliko dugo koliko traje i naše interesovanje za tehniku a to je kod Jugoslovena neiscrpno (Zarazno se nasmejao.) Sad sam ga malo Preterao! (Smeje se i dalje) Ali, ja stvarno mislim da je to tako

Ovaj intervju sa Čedom Mrazom uređen je neposredno po njegovom dolasku u Beograd. Elegantan ljubazan i predusretljiv Mraz je ostavio utisak prijatnog sagovornika odmerenog i promišljenog u odgovorima

*Vesna Ćosić*

Memorije

# Brzo, brže... keš

**Ne tako davno, keš (CACHE) memorija su bile neophodna samo na tržištu računara vrhunskih performansi i samo u ograničenom broju aplikacija. Danas, sa napretkom brzih 32-bitnih mikroprocesora, potreba za keš tehnikama prodire i u oblast proizvodnja sistema namenjenih znatno širem krugu korisnika, uključujući čak i sferu personalnih računara. Šta je keš, kako se koristi i šta se njime dobija?**

Inženjeri koji koriste nove moćne 32-bitne mikroprocesore spoznali su da sistem visokih performansi zahteva mnogo više od snažnog centralnog procesora. Zbog toga su bili prinuđeni da prihvate mnoge tehnike i finese karakteristične za superminiračunare i mejnfrejm računare. Između ostalog, bilo je neophodno usvojiti koncept hijerarhijske organizacije memorije po kojem se između CPU (centralne procesorske jedinice) i glavne memorije smešta brza keš (cache) memorija, čiji je osnovni zadatak da iz „udaljene" memorije stalno prinosi procesoru potrebne podatke na dohvat ruke.

## Za tango je potrebno dvoje

Pojavom 32-bitnih mikroprocesora usko grlo u sistemima visokih performansi postale su memorije. To se jednostavno vidi na sledećem primeru. Zamislite Intel 80386 koji radi na „skromnih" 20MHz (danas već ogromna većina 32-bitnih mikroprocesora radi sa taktom od, barem, 20MHz; chip-set Clipper C300 ima osnovni takt od 50MHz!). 80386 zahteva minimum dva takta da obavi jedan ciklus (čitanje/upis) po magistrali, što iznosi 100 nanosekundi. Jeftine **dinamičke RAM (DRAM)** memorijske komponente nisu dovoljno brze da okončaju ciklus u ovih 100 ns. Rešenje je (naizgled) u korišćenju znatno bržih **statičkih RAM (SRAM)** memorija. Tu, međutim, iskrsavaju veliki problemi. SRAM komponente jesu brze, ali su, istovremeno, i znatno skuplje i imaju manju gustinu pakovanja, pa zauzimaju znatno više površine nego DRAM memorije.

Spore memorije mogu da izuzetno degradiraju performanse centralnog procesora („za tango je potrebno dvoje!"). U već pomenutom primeru sa 80386 ubacivanje jednog neaktivnog ciklusa produžava vreme pristupa podatku za 50%. Zahvaljujući internoj protočnoj (pipeline) organizaciji 80386 (čime se omogućava paralelizam u radu osnovnih blokova procesora) performanse sistema se ne pogoršavaju baš za svih 50% (stvarna vrednost je između 8% i 30%). Moćni procesor u okovima sporih (čitaj jeftinih) memorija? Tužno!

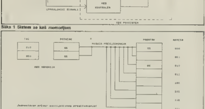
U sistemima sa keš memorijom svi podaci su smešteni u glavnoj memoriji, a neki od njih su duplirani u brzoj keš memoriji. Kada pristupa memoriji, procesor prvo pretražuje keš. Ako se tamo nalazi traženi podatak (pogodak, „cache hit") procesor mu može pristupiti brzo, bez ubacivanja neaktivnih ciklusa („wait" stanja). Ako se traženi podatak ne nalazi u keš memoriji (promašaj, „cache miss") CPU mora pristupiti glavnoj memoriji. Ukoliko su potrebni podaci i program najčešće prisutni u keš memoriji (faktor pogotka „hit rate" veći od recimo 90%) efektivno vreme pristupa glavnoj memoriji se znatno smanjuje.

Sistem sa keš memorijom čine brza SRAM memorije malog kapaciteta i spora DRAM memorija velikog kapaciteta (slika 1). Uz dobar algoritam dupliciranja potrebnih podataka i instrukcijskog koda keš nudi SRAM performanse po ceni koja se približava ceni memorije izgrađene isključivo od DRAM komponenata.

Predviđanje kojoj će sledećoj memorijskoj lokaciji program pristupiti bilo bi nemoguće kada bi programi koristili memoriju potpuno proizvoljno. Na svu sreću, programi obično pristupaju memoriji koja se nalazi u okolini lokacija koje su korišćene u bliskoj prošlosti. U anglosaksonskoj literaturi ovaj princip je poznat pod imenom „locality of reference" ili „program locality".




Slika 1 Sistem sa keš memorijom




Slika 2 Direktno preslikavanje

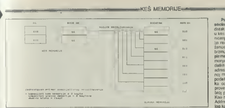
Ova lokalizovanost programa je osnovni preduslov za realizaciju keš sistema. U tipičnim programima izvršavanje naredbi je, najčešće, ili sekvencijalno, ili po kratkim petljama koje obuhvataju bliske memorijske lokacije. Isti podatak se često koristi više puta uzastopno (tj. u kratkom vremenskom intervalu). Programi pristupaju steku uvek u blizini njegovog tekućeg vrha. Nizovi karaktera i vektori često se skeniraju sekvencijalno. Princip „lokalizovanosti" narušavaju programski skokovi ili prelazak na novi posao u multitasking sistemima.

Mada je koncept keš memorije prilično jednostavan, njegova realizacija je složen proces koji obuhvata razmatranje brojnih faktora: načina preslikavanja sadržaja glavne memorije, veličine blokova podataka koji se prenose u keš memoriju, algoritma izmene sadržaja keš memorije, veličine keš memorije i načina ažuriranja sadržaja keš memorije. Nažalost, mnoga iskustva sa mejnfrejm i minikompjuterskog nivoa nisu mogla biti jednostavno preslikana na mikroprocesorske sisteme. Najvažniji razlozi su cena i broj komponenata potrebnih za realizaciju odgovarajućeg keš koncepta.

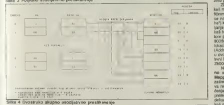
Razmotrimo recimo problem optimalne veličine bloka podataka koji se prenosi između glavne i keš memorije. Suviše mali blokovi mogu biti nedovoljni da obuhvate često korišćene petlje u programima, pa se često zahteva izmene sadržaja keš memorije. Veliki blokovi brzo ispune keš memoriju (koja je skupa i skromnog kapaciteta). To opet zahteva čestu izmenu sadržaja keš memorije. Takođe, sa stanovišta lokalizovanosti programa suviše dugački blokovi nisu potrebni. U mikroprocesorskim sistemima blokovi se sastoje iz znatno manjeg broja bajtova nego što je to slučaj kod velikih računara. Nekad se blok sastoji iz samo četiri bajta, tj. od jedne 32-bitne reči; češći je slučaj da se podržavaju blokovi od 16 bajtova (četiri „duge" reči).

## . . . i keš tehnike

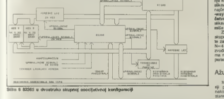
Keš memoriju sačinjavaju dva memorijska područja. U jednom se nalaze podaci koje CPU koristi, dok se u drugom nalaze obeležja tih podataka, tzv. „tag"-ovi (u domaćoj literaturi se često koristi i termin



**Slika 3 Potpuno asocijativno preslikavanje**



**Slika 4 Dvostruko skupno asocijativno preslikavanje**



**Slika 5 82385 u dvostruko skupnoj asocijativnoj konfiguraciji**

„ključ", no po mišljenju autora ovog teksta taj termin je odgovarajući samo za posebnu grupu keš realizacija koje koriste asocijativne memorije) Tag služi za utvrđivanje da li se traženi podatak nalazi u keš memoriji Neki dodatni bitovi mogu se koristiti kako bi se odgovarajući podaci u keš memoriji proglasili nevažećim (tkz. „valid" bit)

Postoje 3 osnovna tipa organizacije keš memorije koji se međusobno razlikuju po načinu preslikavanja („mapping") sadržaja između glavne i keš memorije. To su direktno preslikavanje potpuno asocijativno preslikavanje i skupo asocijativno preslikavanje

Najjednostavniji i najbrži pristup je **direktno preslikavanje (Direct Mapping)** Pri ovoj vrsti preslikavanja memorijska adresa podatka u glavnoj memoriji posmatra se na sledeći način niži biti sačinjavaju **indeks**, a gornji deo adrese je **tag**. Indeksni deo određuje lokaciju u keš memoriji na kojoj se može naći podatak Očigledno samo jedan od podataka koji imaju isti indeksni deo adrese može se u datom trenutku naći u keš memoriji (slika 2) Adresa podatka u cache memoriji jednaka je indeksnom delu adrese podatka u glavnoj memoriji

Kada CPU zatraži pristup određenoj memorijskoj lokaciji prvo se pomoću indeksnog dela adrese pristupi lokaciji u keš memoriji i zatim proveri da li se slažu tag na toj lokaciji i tag (gornji deo adrese) željenog podatka. Ukoliko se slažu CPU koristi podatak iz brze keš memorije

Direktno preslikavanje ima (kao što se može i očekivati) i dobre i loše strane Izvanredne osobine (po kojim direktno preslikavanje nadmašuje ostale tipove organizacije) su jednostavnost i brzina preslikavanja minimalan broj bita potreban za smeštaj taga i neophodnost samo jedne komparacije (radi utvrđivanja da li se podatak nalazi u keš memoriji) Mana je samo jedna (ali izuzetno velika) Naime, preslikavanje je prilično kruto i matrikavno (ne mogu se u keš memoriji istovremeno naći dva podatka sa istim indeksom) što znatno smanjuje faktor pogotka Ipak, u sistemima gde je cena keša glavno ograničenje i uslov pri realizaciji direktno preslikavanje je našlo primenu

**Potpuno asocijativno preslikavanje (Fully Associative Mapping)** omogucava da se bilo koji podatak iz glavne memorije nađe na bilo kojoj lokaciji u keš memoriji (slika 3) Nema dakle nikakvih ograničenja. Sa stanovišta faktora pogotka ovo je najbolja realizacija keš podsistema Međutim cena i složenost realizacije (koja utiče između ostalog i na brzinu) predstavljaju dve velike nepovoljnosti u implementaciji ovakvog načina organizovanja keš memorije Problem je što se u keš memoriji pored podatka mora nalaziti (kao tag) i njegova potpuna adresa (tačnije u praksi skoro cela adresa) u glavnoj memoriji Kada procesor želi da pristupi nekom podatku prvo treba proveriti da li adresa tog podatka odgovara nekom tagu iz keš memorije Treba proveriti sve lokacije u keš memoriji vidi se da je broj potrebnih komparacija jednak kapacitetu keša Kao memorije se koriste asocijativne CAM (Content Addressable Memory) memorije koje su skupe i malog kapaciteta s obzirom da treba ostvariti veliku brzinu pretraživanja

Zbog navedenih razloga potpuno asocijativne keš memorije realizuju se uglavnom, samo u specifičnim oblastima primena mikroprocesora Najčešće se nalaze u sastavu mikroprocesora („on-chip cache") koji u sebi sadrže integrisanu jedinicu za upravljanje memorijom (MMU) Tu se potpuno asocijativni keš koristi za realizaciju labele koje sadrže deskriptore poslednjih korišćenih memorijskih stranica Intel 80386 ima TLB (Translation Lookaside Buffer) sa 32 lokacije a Motorola 68030 ima 22 ulaza u svoj ATC (Address Translation Cache) Jedini mikroprocesor u ovom trenutku koji ima ugrađen potpuno asocijativni keš za instrukcije i podatke (po 256 bajtova) je Z80000

Najčešće se koristi kompromisno rešenje **skupno asocijativno preslikavanje (Set Associative Mapping)**. Keš memorija se deli na N blokova, a zatim se između glavne memorije i pojedinih blokova realizuje direktno preslikavanje (slika 4) Ograničenja pri smeštaju podataka u keš postoje ali nisu preterano striktna Naime podatak se određenim indeksnim delom adrese može se naći na N mesta u keš memoriji Štaviše svi elementi iz glavne memorije koji imaju isti indeks potpuno se asocijativno preslikavaju na N mogućih lokacija. U praksi broj N je najčešće mali Za N=2 u pitanju je **dvostruko (two-way)** skupno asocijativno preslikavanje a za N=4 **četvorostruko (four-way)** skupno asocijativno preslikavanje Ukoliko je N=1 preslikavanje se transformiše u direktno

Zbog manjih restrikcija prilikom preslikavanja skupno asocijativno preslikavanje daje bolje rezultate za faktor pogotka od direktnog preslikavanja (za N=4 napredak je i do 40% prema podacima proizvođača odgovarajućih keš kontrolera) Brzina sistema nije znatno ugrožena jer je broj potrebnih komparacija relativno mali (jednak je broju N)

## Ažuriranje memorije

Kada se traženi podatak ne nalazi u keš memoriji potrebno ga je preneti u keš i prepisati preko nekog podatka (odnosno bloka podataka) koji je zastareo i neaktuelan. Pri tome se koriste različiti kriterijumi aktuelnosti slučajni kriterijum FIFO kriterijum (neaktuelan je onaj podatak koji se najduže nalazi u keš memoriji) **LRU** kriterijum (**Least Recently Used**, među podacima u kešu neaktuelan je onaj podatak koji je najranije korišćen) i LFU kriterijum (Least Frequently Used, najređe korišćen podatak je neaktuelan) Najčešće se koristi LRU kriterijum (koji daje dobar faktor pogotka uz relativno jednostavnu realizaciju) i većina savremenih integrisanih keš kontrolera ima u sebi ugrađen LRU algoritam.

U keš sistemima dve kopije istog podatka mogu istovremeno postojati jedna u glavnoj a druga u keš memoriji Zbog toga keš mora imati ugrađen sistem za ažuriranje kako bi se sprečilo postojanje i korišćenje zastarelih podataka („stale data") To se postiže na različite načine U **„write through"** sistemu keš kontroler istovremeno upisuje podatak i u keš i u glavnu memoriju. Ovakav pristup je jednostavan i nije sklon greškama Performanse sistema su međutim neke zbog vremena potrebnog da se podatak izmeni u glavnoj memoriji kao i zbog „gustog saobraćaja" prema glavnoj memoriji (što može biti

od izuzetnog značaja u multiprocesorskim sistemima) Bolje rezultate omogućava **baferovani „write through"** po koma se podatak istovremeno upisuje u keš memoriju i „write" bafer kontrolera (a ne u sporu glavnu memoriju) Podatak se zatim nezavisno od CPU prenosi u glavnu memoriju Hardver je nešto komplikovaniji, ali je procesor oslobođan nepotrebnog čekanja prilikom svakog upisa u keš. Naravno ukoliko postoji samo jedan „write" bafer dva uzastopna procesorska ciklusa upisa neteraće CPU da čeka Takođe, ciklus upisa praćen ciklusom čitenja koji se završio promašajem primoraće procesor da čeka. U **„write-back"** sistemu tag odgovarajućeg bloka podataka u keš memoriji sadrži i tkz „bit izmene" Ovaj bit se setuje ako je u bloku došlo do izmene nekih podataka tako da cache i glavne memorija ne čuvaju ekvivalentne sadržaje Kada namesto nekog bloka u keš treba doneti novi blok podataka prvo se testira „bit izmene" bloka koji je proglašen neaktuelnim Ukoliko je pomenuti bit setovan blok podataka se iz keš memorije prepiše u glavnu memoriju pre nego što se keš memorija napuni novim aktuelnim sadržajem

Posebni problemi sa očuvanjem koherentnosti keš memorije nastaju u sistemima gde više od jednog uređaja ima pristup glavnoj memoriji (multiprocesorski sistemi ili sistemi sa DMA na primer) Zamislite slučaj kada jedan uređaj upisuje podatke u glavnu memoriju a drugi uređaj istovremeno koristi na taj način već zastarele podatke koji se nalaze u njegovoj keš memoriji Rešavanja ovog i sličnih problema daleko prevazilazi okvire ovog članka Primer je naveden kako bi se još jednom istakla sva kompleksnost sistema sa keš memorijom

## Keš kontroleri

Tri su osnovna pravca razvoja sisteme sa keš memorijom

— realizacija sistema koji sadrža integrisane keš kontrolerske čipova (Intel 82385 Chips & Technologies 82C312 Austek A38152 NEC μPD43606 National Semiconductor NS32605)

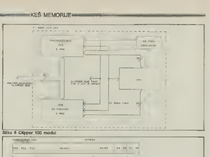
— realizacija sistema u kojima su keš kontroler i memorija realizovani u čipu (najčešće je i MMU integrisana u istom čipu što ce biti naknadno objašnjeno) koji je sastavni deo procesorskog modula izgrađenog iz nekoliko čipova visokog stepena integracija (čip-set Clipper AT&T 32200/32201)

— realizacija keš memorije (instrukcijski keš i keš za podatke) u okviru 32-bitnih mikroprocesora (Motorola 68030 Z80000 NS32532 NEC V70)

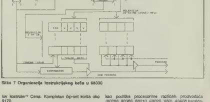
Tipičan keš kontroler se sastoji iz nekoliko osnovnih celina. To su RAM u koji se smeštaju adresni tagovi komparatori adresa i kontrolna logika za upravljanja keš memorijom koja se najčešće nalazi izvan kontrolerskog čipa (NEC je odstupio od ovakve organizacije integrišući kontroler i memoriju na jednom čipu)

**Intel 82385 omogućava da se 32K eksterne keš memorije mapira pomoću direktnog ili dvostrukog skupno asocijativnog preslikavanja. U čipu su realizovani LRU algoritam, kao i baferovani upis. Na slici 5 predstavljen je jedan tipičan keš sistem zasnovan na 80386 i 82385 82385 kontroliše dva bloka keš memorije, kao i protok adresa i podataka prema sistemskoj magistrali preko koje 80386 ima pristup glavnoj memoriji 82385 ima i ugrađenu logiku za nadgledanja magistrala („bus watching", „bus snooping") Na taj način se obezbeđuje očuvanje koherentnosti keš memorije u multi-master sistemima. 82385 može da adresira svih 4Gb glavne memorije (kao i 80386) Cena? Sitnica! Ako kupujete na velike, između $130 i $160.**

**82C312 (Chips&Technologies)** ima dosta zajedničkih osobina sa 82385 dvostruko skupno asocijativno preslikavanje podrška do 32K keš memorije ugrađen LRU algoritam mogućnost baferovanog upisa u glavnu memoriju Logika za nadgledanje magistrala nije ugrađena i mora se eksterno realizovati kako bi se sprečilo da procesor pristupa nevažećim podacima u glavnoj memoriji Takođe kontroler može da adresira samo 64M glavne memorija 82C312 je deo „motherboard" čip-seta CS8231 i sadrži u sebi integrisan i DRAM kontroler za glavnu memoriju. Koja je glavna prednost u odnosu na Intelov kontroler? Cena. Kompletan čip-set košta oko $170



Slika 6 Clipper 100 modul



Slika 7 Organizacija instrukcijskog keša u 68030

Australijska firma **Austec Microsystems** je prošle godine izbacila kontroler **A38152** koji po mnogo čemu nadmašuje predhodna dva (lako je uočiti da se nastavija tradicija izvanredne podrške Intelovih procesora) Da bi se povećao faktor pogotka uvedena je četvorostruka skupno asocijativna organizacija (Austek garantuje poboljšanja od 25% u odnosu na 2 way organizaciju odnosno 40% u odnosu na direktno preslikavanje) Ovo se obrazlaže činjenicom da memorijski zahtevi 80386 imaju barem tri izvora instrukcijski segment stak i najmanje jedan data segment Pri jednostavnijoj organizaciji keš memorije naznačena tri zahteva za memorijom su često u konfliktu što obara faktor pogotka A38152 podržava do 32K keš memorije i obezbeđuje nadgledanje magistrale u cilju održavanja koherentnosti keš memorijskog sistema A38152 omogućava rad 80386 bez neaktivnih ciklusa sa 45ns SRAM memorijama (16MHz sistem) i 35ns SRAM memorijama (20MHZ sistem) U sastavu kontrolera je naravno i „write" bafer Specifičnost Austekovog čipa je da se programski mogu definisati dve memorijske oblasti (veličina između 64Kb i 4Gb) čiji se sadržaji ne smeju premeštati u keš (nema smisla memorijski mapirane ulazno/izlazne uređaja „premeštati" u SRAM memoriju) Algoritam izmene sadržaja u keš memoriji je slučajan Austekovi stručnjaci tvrde da bi ugradnja LRU algoritma „pojela" suviše silicijuma a da se zahvaljujući četvorostrukom skupno asocijativnom preslikavanju razlika ne može ni primetiti A38152 je gotovo neverovatno jeftin $55 (u količinama preko 10000)

**NEC**-ov čip μPD43608 je zamišljen kao univerzalna, veoma programabilna komponenta koja se može koristiti i u 16-bitnim i u 32-bitnim sistemima, kao podrška procesorima različitih proizvođača (80286 80386 68010 68020 V60) 43608 karakteriše još nekoliko osobina po kojima se izdvaja od do sada opisanih kontrolera između ostalog na čipu se nalazi integrisana i sama keš memorija (kapaciteta 8Kb) Veći kapacitet keš memorije i bolji faktor pogotka se mogu postići vezivanjem do četiri 43608 paralelno Adresni prostor kontrolera je 256Mb LRU algoritam je ugrađen na čip koji se sastoji od 620 000 tranzistora (80386 ima u sebi 275.000 tranzistora, a 68020 oko 200 000)!

Još jedan veliki proizvođač je najavio svoj keš kontroler semplovi će biti raspoloživi krajem godine To je **National Semiconductor NS32605** Nije univerzalan kao NEC kontroler (namenjen je sistemima sa procesorom NS32532), ali može da adresira najveći keš (do 256K) Opcionalno, u sistemu mogu postojati dva ovakva kontrolera i tako realizovati razdvojeni instrukcijski keš i keš za podatke Kontroler će podržavati direktno preslikavanja i dvostruko skupno asocijativno preslikavanje.

## Keš u procesoru

**Poseban trend** u razvoju keš sistema su keš realizacije blisko povezane sa centralnim procesorima. Keš kontroler i memorija se integrišu u jedan čip i postaju sastavni delovi procesorskih modula sastavljenih od nekoliko čipova Neki proizvođači keš (istine manjeg kapaciteta zasada najveći keš ima NEC V70—2K) realizuju u okviru mikroprocesora Postoje znatne razlike u stavu prema keš memoriji implementiranoj u mikroprocesoru Intel je recimo odlučio da sačeka sa uvođenjem keša na svoje mikroprocesore sve dok razvoj tehnologija ne omogući integraciju keš memorije većeg kapaciteta

**AT&T** je keš podsistem **WE32201** realizovao u okviru svoje 32-bitne familije koju sačinjavaju mikroprocesor WE32200, matematički akcelerator WE32206, DRAM kontroler WE32103, DMA kontroler 32104 i generator takta WE32202. WE32201 sadrži u sebi 4K brze memorije, „bus watching" logiku, ima ugrađeno dvostruko skupno asocijativno preslikavanje i obezbeđuje faktor pogotka od 85% pri radu centralnog procesora bez neaktivnih ciklusa, na 30MHz. Kao i NEC-ov kontroler i WE32201 se može kaskadirati radi ostvarivanja većeg kapaciteta memorije i boljeg faktora pogotka. Posebno je interesantno da je keš integrisan na istom čipu sa jedinicom za upravljanje memorijom (MMU). Da bi se razumeli razlozi za ovakvu koncepciju treba prvo objasniti pojmove virtuelnog i fizičkog keša. **Virtuelni keš** se nalazi između CPU i MMU i radi sa virtuelnim adresama. U slučaju pogotka nije potrebno prevođenje virtuelnih u fizičke adrese od strane MMU. Ovakav sistem omogućuje veliku brzinu rada, ali prouzrokuje velike probleme u multi-master sistemima (da li je potrebno ponavljati priču o koherentnosti keš sistema). **Fizički keš**, kao što i sam naziv kaže, radi sa fizičkim adresama i u takvim sistemima može biti velikih problema sa brzinom rada jer je potrebno prevođenje adresa pre svakog pristupa keš memoriji. Ažuriranje je međutim jednostavnije. Kombinovanjem kasa i MMU na istom čipu u velikoj se meri eliminiše problem kašnjenja u sistemima sa fizičkim kešom.

Sličan koncept realizovao I **Fairchild** (kasnije je prava prodao **Intergraph**-u) svojim čip-setom **Clipper** (slika 6). Clipper modul sačinjavaju 3 čipa: RISC CPU sa ugrađenom floating point izvršnom jedinicom, instrukcijski keš/MMU i keš za podatke/MMU (postoje dve verzije modula: C100 za 33.3MHz i C300 za 50MHz). Svaki keš je kapaciteta 4K. Faktor pogotka je preko 90% (96% za instrukcijski keš), što je, između ostalog, rezultat dvostrukog skupno asocijativnog preslikavanja. Efektivno vreme pristupa memoriji u Clipper sistemu je oko 108ns (za 33.3MHz verziju).

**National** je odlučio da u svoj najjači 32-bitni mikroprocesor **NS32532** ugradi 1024 bajta fizičkog keša za podatke i 512 bajta fizičkog keša za instrukcije. Zanimljiva je različita organizacija ove dve keš memorije: instrukcijski keš je sa direktnim preslikavanjem, a keš za podatke ima dvostruku skupno asocijativnu organizaciju. Naime, instrukcijski niz je, manje-više, sekvencijalan i direktno preslikavanje daje sasvim dobar faktor pogotka za instrukcijski keš. Strukture podataka su znatno razuđenije, pa su National-ovi stručnjaci smatrali da keš za podatke zahteva složeniju organizaciju.

Procesor **Z80000** je specifičan po tome što je svoja dva 256-bajtna keša (za instrukcije i za podatke) organizovao potpuno asocijativno. Da bi se smanjio broj potrebnih komparacija, keš je organizovan kao 16 linija (ulaza) po 4 „duge" 32-bitne reči. Tako se keš sastoji od 28-bitnog adresnog tag-a, podatka i bita koji označava aktuelnost podatka u memoriji (da li se slažu sadržaj glavne i keš memorije). LRU algoritam je implementiran preko steka. Keš „pogodak" zauzima mesto na vrhu steka. Kada prilikom ciklusa očitavanja dođe do promašaja, dno steka se gubi, a na vrh steka dolazi obeležje traženog podatka koji se prebacuje u keš. Kada se dogodi promašaj prilikom ciklusa upisa, LRU stek se ne menja. Samo se izmeni podatak u glavnoj memoriji.

**Motorola** je ugradila 256-bajtni instrukcijski keš već na 68020. Naredni procesor u nizu, **MC68030**, je pored instrukcijskog integrisao i keš za podatke i MMU. To je znatno unapredilo performanse Motorolinih mikroprocesora. Integrisani keš je smanjio potrebu za eksternim aktivnostima procesora i znatno skratio vreme pristupa memoriji. Tkz. „burst" mod prenosa podataka (4 „duge" 32-bitne reči u 4 taktna ciklusa) omogućava da se podaci već nalaze u brzoj keš memoriji kada se prvi put ukaže potreba za njima. Najvažnija činjenica je da su instrukcijski i data keš autonomni i da im se može simultano pristupati. Naime, u 68030 ostvaren je visok stepen paralelizma. Moguće je istovremeno očitati instrukciju iz instrukcijskog keša, potreban podatak iz keša za podatke i obaviti nekakav eksterni pristup određenom periferijskom uređaju (MC68030, štaviše, omogućava da se istovremeno odvijaju interne instrukcije i instrukcije koje zahtevaju eksterne cikluse).



Instrukcijski keš je direktno mapiran 256-bajtni memorijski prostor koji sačinjavaju 16 blokova od po 4 32-bitne reči. Način selekcije podatka u keš memoriji je prikazan na slici 7. Gornjih 24 bita adrese sačinjavaju tag bloka i koriste se pri utvrđivanju da li se traženi podatak nalazi u kešu. Kao indeksni deo kojim se vrši selekcija bloka koriste se biti A4 do A7. Status FC2 definiše privilegije korisnika keša (user/supervisor). Biti A3 i A2 selektuju 32-bitnu reč i odgovarajući „valid" bit. Kada se naredna instrukcija ne pronađe u kešu (recimo, u programu je usledio nekakav dug skok) keš se automatski puni potrebnim instrukcijskim kodom.

Keš za podatke je organizovan na veoma sličan način (razlika je u tome što u tag ulaze sva tri statusa FC0-2 koji definišu adresni prostor koji odgovara bloku podataka u keš memoriji). Ukoliko se pri ciklusu upisa podatak pronađe u keš memoriji, ažuriraju se obadve njegove kopije u glavnoj i keš memoriji (write-through). Ukoliko se podatak ne pronađe u brzoj memoriji, menja se samo sadržaj glavne memorije.

Programer ima jako mali uticaj na funkcionisanje keš memorija u 68030. Omogućene su samo sledeće aktivnosti: brisanje keša ili pojedinog njegovog elementa, kao i isključenje celokupnog keš mehanizma (hardverski se to može postići preko pina CDIS/).

## Može li jeftinije?

Sistemi sa keš memorijom su još uvek veoma skupi. U mnogim aplikacijama razvojni inžinjeri su voljni (ili prinuđeni) da donekle žrtvuju performanse sistema kako bi njihov dizajn imao razumnu cenu. Veliki proizvođači misle i na to! U okviru Chips-ovog AT čip-seta za podršku 80386 nalazi se integrisani memorijski kontroler 82C302 koji omogućava da sistem bez keš memorije očuva solidne performanse i bez korišćenja brzih SRAM komponenata. Ideja je sledeća: DRAM memorije zahtevaju dve komponente adrese: adresu vrste (Row Address, koja se čuva u odvojenom baferu memorijskog čipa) i adresu kolone (Coloumn Address). Ako DRAM čipu nije potrebno dovesti prvu komponentu adrese (tkz. stranični „page" mod rada) uštedi se jedan (tkz. RAS, Row Address Strobe) ciklus i memorijski pristup može biti brži. Pomenuti memorijski kontroler koristi to svojstvo. Ukoliko je adresa koju na magistralu izbaci procesor u okviru određenog bloka adrese (dužina 2K), memorijama se dovodi samo adresa kolone. Tako se zbog lokalizovanosti podataka postižu (statistički gledano) memorijski ciklusi sa manje od 0.5 (Chips-ovi stručnjaci operišu sa cifrom od 0.3) neaktivnih ciklusa po jednom pristupu. Većina savremenih DRAM kontrolera na sličan način koristi osobinu da DRAM memorije mogu da rade u pomenutom „straničnom" modu.

*Ivan Radivojević*

## Nonsense in Basic

# Dobitak gubitka

Ako neki čitalac „Računara" uspe da odgovori na pitanje koje me muči ovih dana, javno izjavljujem i potpisujem da ću mu platiti ručak u nekom od najboljih beogradskih restorana.

Ali napre da vam ispričam o čemu je reč, pitanje ću ostaviti za kraj.

Beogradska lutrija je nedavno uvela nove elektronske mašine za uplatu loto. Obradovao sam se, igram loto pa sam se ponadao da ni ja ni službenica na uplatnom mestu nećemo više gubiti vreme dok ona okreće ručicu stare mehaničke mašine, odavno zrele za muzej prethodne tehnološke ere. Kad ne lezi vraže! Ispred kioska za uplatu otegao se red kao suša osamdeset osme.

Pomiren sa sudbinom, stao sam na začelje. Sat i deset minuta kasnije pružio sam kroz otvor šaltera dva tiketa. Službenica je u mašinu ubacila prvi. Začuo se neki romor, žvakanje, zujanje.

— Šta se to dešava? — pitao sam.

Uzdahnula je kao da se nikad neće udati.

— Ništa, komšija. Jedino mi sad trebe trideset puta više vremena nego sa starom mašinom.

Mašina prestade da grgoće.

— Gotovo! — sa olakšanjem je rekla službenica. — Ovaj tiket je u redu.

Upravo sam hteo da primetim da je ono „trideset" možda preterano, jer je reč bio samo sedam ili osam puta duži nego obično, kad mašina kod drugog tiketa beznadežno zakrklje.

— Popunite novi tiket, molim vas — reče službenica.

— A šta s ovim nije u redu? — pitao sam. — Popunio sam ga isto kao i prvi.

— To niko ne zna. Mašina ne prima.

Da ne dužim, situacija je takva da, čak i kad mašine prima, igračima za uplatu treba nekoliko puta više vremena nego ranije. A kad ne prime... bolje je ne pominjati.

Tako je Beogradska lutrija uvela automatizaciju i dobila — gubitak vremena. Učinak je očigledno kontraproduktivan.

Moje pitanje glasi: kako je to moglo da se dogodi?

Sledeće odgovore odbacujem kao nelogične, nemoguće ili maliciozne:

- Projekt automatizacije su radili nestručnjaci.
- Stručnjacima ili ljudima iz Lutrije nije palo na pamet da uporede vreme uplate sa starim i novim mašinama.
- Proizvođač ili nosilac projekta uvalio je naivnoj Lutriji zastarelu i neodgovarajuću opremu.
- Lutrija je uvodila automatizaciju svesno prihvatajući smanjenje produktivnosti uplatnih mesta i otežavanje uplate igračima.
- Niko od partnera u ovom poslu ne zna kako se uplata lota automatizuje u belom svetu.
- Neko je uzeo pare.

Pa šta je onda pravi odgovor? Pomozite mi ostajem pri obećanju za ručak.

*Bata Bajt*

# Ploteri pišu perom

**Crtež na papiru je krajnji cilj. Da bi se posao završio nije dovoljno imati dobar i brz kompjuter, monitor visoke rezolucije i kvalitetan grafički program. Neophodan je i uređaj koji će sve što je urađeno precizno preneti na papir — tušem ili specijalnim flomasterom, u više boja ako tako želimo, bolje i mnogo brže od najboljeg tehničkog crtača, a bez grešaka. Jednom rečju potreban je — ploter.**

Prvi ploter za komercijalno tržište napravljen je 1958 godine u tada anonimnoj američkoj kompaniji CALCOMP (CALifornia COMputer Products) u Anahajmu, u blizini Los Anđelesa

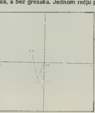
Ličio je pomalo na ručni geštatner ali bez ručice za okretanje Valjak (engl drum) prečnika oko 18 cm i širine oko 30 cm okretao se napred nazad a iznad gornjeg dela valjka bila je poprečno smeštena šina po kojoj se pomoću tanke sajle levo-desno kretala glava sa jednim perom Papir perforiran kao kod printera dolazio je sa rolne iza valjka prelazio preko valjka i namotavao se na drugu rolnu ispred valjka Valjak predstavlja X osu, a glava sa perom Y osu Obe ose su pokretane step motorima Elektronski deo je bio izrađen kompletno u tehnici germanijumovih tranzistora

Za današnje pojmove ovaj ploter je bio izuzetno spor ali je — crtao Naravno kvalitet crteža nije bio ni na nivou današnjih jeftinih modela ali za to davno vreme je bio to veliki uspeh Poređenja radi možemo reći da je crtež bio približno jednak slici koja se dobije na IBM-PC kompjuteru uz korišćenje EGA kartice i monitora

Veliki broj ovih prvih modela plotera napravljen je za kompaniju IBM koja ih je prodavala uz tadašnji kompjuter IBM 3 model koji je bio vrlo popularan više godina a čak i danas se može ponegde naći u upotrebi (zajedno sa ploterom)

## Komforan rad

Danas je u upotrebi više vrsta plotera Najbrojniji su oni koji crtaju perom ili kako bi Amerikanci rekli pen-ploteri Pod perom se podrazumeva hemijska olovka flomaster, rapidograf sa tušem ili neka varijanta ovog pisaćeg pribora Ovi ploteri crtaju na ravnom papiru pausu ili plastičnoj foliji Proizvo



Primer kako ploter koristi relativne koordinate




da se od formata A4 do A0, a ima i većih Pen-ploteri se često nazivaju i vektor ploteri, zbog toga što pero podvlači liniju, vektor kao rezultatu kretanja X i Y ose

Opis prvog komercijalnog plotera što se mehanike tiče, po ideji odgovara i današnjim savremenim ploterima Princip sa valjkom i šinom se zadržao do današnjih dana uz mnoga poboljšanja Danas se za pokretanje valjka i glave sa perima koriste brzi i precizni istosmerni servo motori umesto step-motora, koji su spori nedovoljno precizni i bučni Elektronika je doživela mnogo veće promene pojavom integralnih kola i mikroprocesora, koji su učinili da savremeni ploteri često imaju moćniji mikroprocesor od kompjutera na koji su vezani

Ugrađeni softver (firmware) omogućuje veoma komforan rad sa ploterom Tu su razni fontovi slova, generatori kružnih i crtkanih linija merenje dimenzija papira na kome se crta prepoznavanje vrsta pera i prilagođavanje svih parametara za to pero, smanjivanje i uvećavanje crteža, crtanje „prozora" (engl windowing) rotiranje crteža i još dosta toga (o čemu će biti više reči kasnije)

Veliki bafen prihvataju često i kompletan crtež pa se uz veće brzine crtanja više ne događa da ploter dugo „zadržava" kompjuter već se pogotovu na sistemima sa više korisnika redovno dešava da ploter čeka na kompjuter Ovo je značajan podatak ako znamo da su prvi ploteri crtali po nekoliko sati crtež koji savremeni ploter uradi za 10–15 minuta

Pošto je brzina crtanja neuporedivo veća — ima plotera koji povlače liniju brzinom od 75 cm u sekundi — bilo je neophodno, na primer napraviti komore u kojima vakuum drži papir čvrsto uz valjak da se ne podiže pri velikoj brzini Ove komore su ujedno jedna vrsta „bafera" za papir Dugo se koristio isključivo papir u rolni To ima prednosti jer je vođenje papira besprekorno mogu da se crtaju i dosta dugi crteži ali i mana jer treba seći papir i skida ti rolnu pa onda ponovo postavljati papir ako se crta samo po jedan crtež Ako se radi tušem onda treba paziti da se kod dužih linija koje ploter povlači tuš osuši pre nego što se papir namota na rolnu Zbog ovog na izgled sitnog problema razvijen je i tuš koji se brzo suši

Ali najveća mana ovog papira u rolni svakako nije problem sušenja tuša Problem je u tome što je ovakav papir dosta skup najviše zato što perforacija koju ima celom dužinom sa oba kraja mora da bude izvedena maksimalno precizno To se radi na specijalno prilagođenim mašinama uz strogu kontrolu Odstupanje u perforaciji koja je osnova za vođenje plotera po X osi odmah bi se odrazilo na tačnost crteža I sastav papira je neobično važan Konsti se papir pravljen na bazi cmogoričnih vlakana i pamuka što obezbeđuje da bude odlična podloga za crtanje a i da se ne širi i ne skuplja previše kod promena vlažnosti

## Problemi i rešenja

Kompanija „Hewlett Packard" je doskočila ovom problemu tako što je modifikovala princip sa valjkom tako da može da koristi komade papira bez perforacije proizvoljne veličine do maksimalnog formata. To je izvedeno na taj način što su iznad levog i desnog kraja valjka postavljeni mali točkići od tvrde gume Površina valjka koju točkići dodiruju je prevučena slojem brusnog kamena nalik na šmirgl-papir Kada se komad papira ubaci između ovih točkića i valjka oni ga pritisnu i utisnu trag po ivicama papira tako da ga nepogrešivo vode napred nazad Ovo rešenje danas primenjuju gotovo svi proizvođači kako na malim tako i na velikim ploterima A0 formata

Postoje i ploteri koji mogu da rade i sa komadima papira (engl cut sheet) i sa papirom u rolni Proizvođači ih reklamiraju pod nazivom „dual mode" ploteri, i oni su hit na tržištu plotera, ali i znatno su skuplji Kombinacija komad papira/rolna je u stvari odličan spoj dobrih osobina oba vanjenta Kod ovakvih modela može se koristiti komadni papir za probne i pojedinačne crteže a rolna za seriju i finalne crteže Zanimljivo je pomenuti da je rezolucija kod

## Šta je ploter

Ploter je elektronsko-mehanički uređaj sa mogućnošću kontrolisanog pomaka po koordinatnom sistemu koji se sastoji od X i Y ose u čijem preseku se nalazi glava sa perom Pomoću interfejse povezan je sa kompjuterom Digitalne signale primljene od kompjutera pretvara u analogne kojima će pokrenuti motore za X ili Y osu i spustiti odnosno podići pero

Jasno je da se u odgovarajućoj sinhronizaciji ova dva kretanja i spuštanjem i dizanjem pera može nacrtati bilo kakav crtež

Naravno potrebno je da se u kompjuteru nalazi odgovarajući program da bismo na papiru dobili grafičke rezultate nekog projekta ili proračuna

Doslovno prevedena sa engleskog reč „plotter" znači začetnik zabeležiti zacrtati uneti (u geografsku kartu) Na našem jeziku bismo mogli da kažemo „crtač" ili „kompjuterski crtač" ali reč ploter je već ušla u domaći kompjuterski rečnik pa ćemo je tako i prihvatiti u ovom tekstu

stog plotera za nijansu bolja kada radi sa rolnom, zbog toga što papir vođen perforacijom i potpomognut vakuumom odlično prianja uz valjak.

Postoji još jedna varijanta plotera sa valjkom koju je lansirala 1975. godine kompanija tvorac prvog plotera CALCOMP. To je takozvani „belt-bed" ploter koji na originalan način rešava problem skupog papira u rolni, a zadržava sve njegove prednosti. Ideja je u tome da se ispod valjka, na oko metar niže, postavi još jedan valjak. Glavni i pomoćni valjak se spajaju specijalnom folijom od milara čiji je početak spojen sa krajem tako da je i ona u obliku valjka. Time se dobija ravna površina koju pokreće glavni valjak (to jest X osa). Na ovu foliju se jednostavno komadićima selotejpa zalepi papir na koma se crta.

Ovo rešenje se možda čini komplikovanim, ali je izuzetno uspelo i CALCOMP je napravio seriju plotera visokih performansi formata A0 na ovom principu. Danas svoj top model proizvodi na tatom principu i jedino je dodala mogućnost rada sa papirom u rolni. Cena mu je oko 40.000 dolara. Sem CALCOMP-a nijedan drugi proizvođač ne prvi „belt bed" plotere.

U opisivanju pen-plotera nikao ne treba preskočiti ravne (flat-bed) plotere kod kojih se papir za crtanje polaže na ravnu podlogu i nije pokretan. Iznad papira takođe postoji šina koja se cela kreće napred-nazad, a po njoj se levo-desno kreće glava sa osrom. Interesantno je da ravnih plotera gotovo i nema među ploterima srednje kategorije. Uglavnom se proizvode u formatima A4 i A3 i namenjeni su korisnicima PC kompjutera. Ili ih nalazimo među ploterskom elitom kao uređaje sa najboljim performansama što se mogu postići. Prave se u vrlo velikim formatima i koriste u kartografiji, za graviranje i tamo gde je potrebno da sve bude „naj". I cena je „u tom stilu": oko 200.000 dolara. Jedan od čuvenih plotera u ovoj kategoriji je CALCOMP-ov model 748, „state of the art" među ploterima. On može da crta sa četiri pera na formatu 122 x 205 cm, brzinom od 76 cm/s uz rezoluciju od 0.005 mm. Da bi se slika upotpunila, treba još reći i da je težak 635 kg!

## Dijamantom po foliji

Postoje i ploteri koji umesto pera imaju dijamantski nož, kojim graviraju po specijalnoj plastičnoj foliji skidajući neprovidni sloj na gornjem delu folije. Tako se dobija izuzetno precizan pozitiv koji se foto-postupkom može smanjiti do milimetarskih dimenzija zadržavajući tačnost i kontrast. Ova tehnika se koristi u izradi „maski" kod projektovanja integralnih kola.

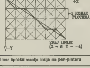
Postupkom graviranja crtaju se i precizne geografske karte, kao i sve što zahteva maksimalnu preciznost. Za ovo se koriste isključivo ravni ploteri većeg formata i visoke preciznosti na kojima se umesto glave sa perima montira posebno konstruisana glava sa dijamantskim nožem.

Ploteri koji „crtaju" svetlom direktno na foto-osetljivoj podlozi (filmu) zovu se foto-ploteri i oni služe kod raznih aplikacija gde se traži visoka preciznost — na primer kod izrade filmova za štampane ploče u elektronici. Upotrebom ovakvog plotera film za izradu štampane ploča dobija se direktno, a preskače se faza crtanja na pen-ploteru i snimanje tog crteža specijalnom kamerom.

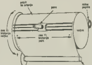
Postoji još jedna vrsta plotera koju treba pomenuti tim pre što je ona u velikoj ekspanziji i po predviđanjima stručnjaka biće još više zastupljena na tržištu: elektrostatički printer-ploteri. Ovi ploteri crtaju na principu sličnom radu foto-kopirnog aparata. I kod njih je primenjen neizbežan valjak, samo što umesto pera iznad valjka celom dužinom poprečno stoji nepokretna glava sa puno segmenata. Glava vrlo precizno specijalni papir koji prolazi ispod naelektriše samo na onim mestima gde nešto treba da bude nacrtano. Zatim se dodaje toner koji ostaje samo na naelektrisanim mestima. Posle prolaska kroz komoru za sušenje crtež je gotov. Sve ove operacije obavljaju se u jednom prolazu papira koji se kreće samo napred. Crta se odjednom cela „štafna" crteža slično linijskom printeru. Ovi printeri su brzi i daju dobar kvalitet crteža. Proizvode se u crno-beloj i kolor varijanti. Zahtevaju moćne kontrolere koji rasterizuju crtež što stiže iz kompjutera u vektorskom obliku, a to im znatno povećava cenu. Koristi se za crtanje rezultata istraživanja u geodeziji i seizmologiji, gde je pogrebno u kratkom vremenu obraditi veliki broj podataka i napraviti crtež. Naravno



Turet sa osam pera plotera „CalComp" 1044




Primer aproksimacije linije na pen-ploteru




Osnovni princip rada plotera sa valjkom

## Mali ploterski rečnik

Kada je reč o ploterima, pored uobičajenih kompjuterskih pojmova nailazimo i na neke koji su vezani samo za ovaj uređaj. Često se dogodi da se pomešaju tačnost i rezolucija plotera, što nikako nije tačno. Evo zbog toga objašnjenja nekoliko najvažnijih pojmova iz ploterske terminologije.

**TAČNOST** *(ACCURACY)*. Rastojanje između tačke gde se vrh pera spustio na podlogu za crtanje posle pojedinačne komande i tačke gde je trebalo da se spusti. Izražena je u inčima ili centimetrima.

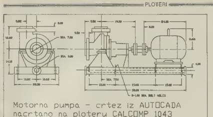
**REZOLUCIJA** *(RESOLUTION)*. Za mehaničke plotere minimalni kontrolisani pomak (korak, step) koji pero može da napravi po X ili Y osi. Za elektrostatičke plotere, broj pojedinačnih tačaka (piksela) koje ploter može da nacrta na jednom inču ili centimetru. Iz ove definicije proizlazi da kod pen plotera cifra kojom se izražava rezolucija treba da bude što manja, a kada se radi o elektrostatičkim ploterima — što veća.

**PONOVLJIVOST** *(REPEATABILITY)*. Mera izražena u inčima ili centimetrima koja pokazuje mogućnost plotera da se vrati na već nacrtanu tačku sa bilo koje druge tačke u okviru formata na kome crta.

**AKSIJALNA BRZINA** *(AXIAL PLOT SPEED)*. Brzina kojom ploter može da povuče pravu liniju po X ili Y osi.

**DIJAGONALNA BRZINA** *(DIAGONAL PLOT SPEED)*. Dijagonalna brzina je 1.414 (koren iz 2) puta veća od aksijalne brzine. Pod dijagonalom se podrazumeva linija koja je 45 stepeni u odnosu na X ili Y osu.

**ISHODIŠTE** *(ORIGIN)*. Tačka iz koje je ploter počeo crtež. Koordinate ove tačke su X=0, Y=0 i one je referentna tačka u odnosu na koju ploter računa broj koraka (stepova).



Motorna pumpa - crtez iz AUTOCADA
nacrtano na ploteru CALCOMP 1043

mogu se upotrebiti i za crtanje klasičnih crteža. Još uvek su dosta skupi — najeftiniji košta oko 50 000 dolara. Međutim, pošto je proizvodnja ovih plotera u ekspanziji, cene polako padaju.

Od svih opisanih plotera jedino se pen-ploteri proizvode i u personalnoj verziji. Zbog toga, a i zato što su ovi ploteri „rodonačelnici", pažnju u ovom tekstu ćemo posvetiti pen ploterima.

## Crtež od osam vektora

Pre nego što krenemo u detaljnije razmatranje kako ploter crta, treba da malo osvežimo znanje iz geometrije o koordinatnim sistemima. Za pen-plotera je bitan Dekartov koordinatni sistem, koji se tako zove po Reneu Dekartu (René Descartes), koji je još u 16. veku otkrio da mesto tačke u ravni u odnosu na bilo koju drugu tačku u istoj ravni može biti određeno matematički koordinatama koje definišu udaljenost te tačke od dve ose označene kao X (horizontalno) i Y (vertikalno).

Dekartov koordinatni sistem je podeljen na četiri kvadranta i svaki kvadrant je definisan pozitivnim ili negativnim vrednostima X i Y. Ako bi se iz tačke A (0 0) povukla linija u bilo kom pravcu, ona bi morala da se nalazi u nekom od četiri kvadranta i svaka tačka na liniji, uključujući i tačku na njenom kraju, mogla bi da bude određena koordinatama. Pošto samo dve tačke definišu celu liniju, proizlazi da kompjuter treba da pošalje ploteru koordinate krajnjih tačaka linija koju treba nacrtati.

Ako usvojimo da je početak svake nove linije koju ploter crta koordinata 0 0, onda je potrebno samo da znamo koordinate kraja linije — i to je jedina informacija koju kompjuter treba da pošalje ploteru, a on će uraditi ostalo. Ovo se bolje razume ako pogledamo skicu 1. Ploter će uvek početi crtež tamo gde smo postavili pero kada smo ga uključili i pripremili za rad. To je koordinata 0 0, ma gde se pero nalazilo. Ako iz kompjutera stigne koordinate X=−5, Y=+2, ploter će nacrtati liniju do te tačke i stati. Za sledeću koordinatu koja stigne iz kompjutera ova tačka je 0 0. Neka nova koordinata bude X1=+8, Y1=+2. Ploter će povući liniju do nove tačke, koja je opet 0 0 u odnosu na sledeću koordinatu. Ovakve koordinate se zovu relativne, za razliku od apsolutnih, koje se računaju od tačke u preseku X i Y ose.

Pretpostavimo da je vrh pera u centru koordinatnog sistema na slici 2, u tački 0 0 i da je pero spušteno. Kada bi se pokrenuo samo motor za Y osu (onaj koji pokreće glavu sa perom levo-desno) i to u levu stranu, vrh pera bi povukao liniju u pravcu označenom sa +Y. Ako bi iz iste tačke 0 0 pokrenuli samo motor za X osu, odnosno valjak u pravcu označenom sa −X, vrh pera bi se „kretao" u pravcu +X. U stvari pero u ovom slučaju stoji, a valjak ispod njega se kreće. Rezultat je linija koja ide ka +X. Ista logika važi i za pravce −Y i −X. U ova četiri pravca kretanje je ostvareno tako što uvek samo jedan motor radi, dok drugi miruje.



Graviranje na filmu uz pomoć dijamantskog noža

## Samostalno delovanje

Ako se oba motora pokrenu u istom trenutku, istom brzinom i za isti broj koraka (stepova), dobićemo linije koje su pod uglom od 45 stepeni u odnosu na ose X i Y. Posmatramo sada situaciju kada valjak ide u pravcu +X a glava u −Y. Rezultat je linija u pravcu +X −Y, odnosno linija pod uglom od 45 stepeni između osa +X i −Y. I za ostale pravce (−X −Y, −X +Y i +X +Y) važi ista analogija.

Ovo su osam osnovnih vektora — pravaca kretanja pera odnosno plotera. To je u stvari maksimalan broj pravaca kretanja pera koje motori na X i Y osi uopšte mogu da naprave, a da linija bude sasvim prava. Sve linije koje se ne poklapaju sa ovim, ploter crta kombinujući osnovne vektore tako da nacrta liniju koja je najpribližnija zadanoj, a sastoji se isključivo od osam osnovnih pravaca. Ovakav postupak se zove aproksimacija.

Mnogi će se sigurno zapitati čemu onda toliko hvale za plotere kada se sve svodi na samo osam vektora koje ploter može da nacrta, a sve ostalo treba aproksimirati. Tajna je u tome što je rezolucija plotera veoma mala, kod boljih modela reda stotog dela milimetra, pa takve linije samo teoretski izgledaju stepenaste, ali u stvarnosti ni lupom se ne mogu videti stepenasti segmenti, jer je već i vrh najtanjeg pera debeo u odnosu na rezoluciju plotera.

U vreme kada su pravljeni prvi ploteri, mogućnosti elektronike nisu dozvoljavale mnogo. Ploteri su mogli da pokreću pero u osam osnovnih pravaca — vektora — da spuste ili podignu pero i to je bilo sve. Sve računske operacije koje su bile potrebne za povlačenje linije (a tu zaista ima dosta da se računa) bile su obavljene u kompjuteru pre nego što bi podaci bili poslati ka ploteru. Fajlovi za crtanje bili su veoma veliki u odnosu na današnje, a trošilo se i puno procesorskog vremena.

Savremeni ploteri su mnogo samostalniji od svojih prethodnika. Obavljaju veći deo posla koji je ranije morao da obavi kompjuter. To smanjuje broj podataka koji idu između kompjutera i plotera i samim tim se smanjuje mogućnost grešaka u prenosu. Takođe, manji je i utrošak procesorskog vremena, što je kod velikih sistema sa više korisnika i te kako značajno. Jedna od najvažnijih računskih operacija plotera je generisanje linije koju treba da nacrta.



## Brojke i slova

To se obavlja u delu koji se zove generator linije (engl. line generator) po algoritmu koji upoređuje zadatu liniju sa najbližim vektorima koji najviše odgovaraju njenom pravcu i tako određuje kretanje plotera. Program koji obavlja generisanje linije nalazi se u ROM-u ili EPROM-u i radi pod kontrolom glavnog mikroprocesora. To je u ovom slučaju savremeni 16-bitni mikroprocesor iz „Motorol'ne" serije 68000.

Primer aproksimacije linije se vidi na slici 3. Zadata linija polazi iz tačke X=0, Y=0 (početna tačka koordinatnog sistema) i ide do tačke X=+6, Y=−4. Prava linija između ovih koordinata predstavlja idealnu zadatu liniju, a izlomljena linija je aproksimacija koju je izračunao i nacrtao ploter. Mreža kvadrata i dijagonala predstavlja moguće pravce kretanja plotera (osam vektora). Nacrtan je na ploteru CALCOMP M84, najmanjem modelu ove kompanije, koji radi crteže formata A4, a ima osam pera i rezoluciju 0.1 mm.

Alfanumerička znake ploter tretira kao i bilo koju drugu liniju. Drugim rečima, svako slovo, broj, simbol ili znak interpunkcije trebalo bi u vidu koordinata poslati iz kompjutera u ploter i on bi ih nacrtao. Pošto ih u svakom crtežu ima dosta, ovakav način bi silno iskomplikovao crtanje i zbog toga je ovaj problem rešen slično kao kod matričnog štampača, gde ASCII kod inicira odgovarajuću kombinaciju iglica.

Za svaki alfanumerički znak je napravljen mali „program", odnosno algoritam kretanja pera, i svi su smešteni u ROM-u. Adrese ovih algoritama su usaglašene sa ASCII tabelom i iz kompjutera treba samo poslati odgovarajući kod koji će pokrenuti crtanje znaka. Od datih parametara zavisi da li će tekst biti crtan pod uglom, vertikalno ili kako je već potrebno. Savremeni ploteri imaju u ROM-u skupove znakova za više jezika, razna pisma, kao i simbole potrebne za crtanje crteža iz raznih struka. Deo plotera koji sadrži alfanumeričke znakove zove se karakter generator.

Pored slova, brojeva i raznih simbola, u firmveru plotera ugrađuju se i algoritmi za crtanje krugova, što pojednostavljuje crtanje kruga — dovoljno je dati koordinate centra i poluprečnik. Kod crtkanih linija (engl. dashed lines) na primer treba samo upotrebiti kod koji će naznačiti ploteru da je linija crtkana i on će sam proračunati broj segmenata srazmerno dužini linije i voditi računa o tome da linija počne i završi crticom, a ne praznim segmentom.

(U sledećem broju ćemo, na nastavak opisa plotera, predstaviti jedan mali „personalni" ploter A4 formata kao i jedan profesionalni model.)

*Bora Milenković*

Akceleratori

# Kako frizirati PC

**U dijalektičkom svetu računarske tehnike niko nije pošteđen brzog napredovanja, pa ni IBM PC XT i ogromna armija njemu kompatibilnih računara. Toga su bili svesni i ljudi iz IBM-a, pa su požurili da novim računarom, koji je dobio ime IBM PC AT, otklone mnoge slabosti ranije arhitektura a time i relativno malu brzinu rada. S druge strane, hiljade vlasnika XT-a pitalo se da li je kvalitativni skok dovoljan da isplati velika ulaganja u novu mašinu. Još dok su se rešavala ova fundamentalna pitanja, i sam AT je postao žrtva napretka. U opštoj nedoumici stvar u ruke preuzimaju male američke firme, uočivši pravi eldorado, a jedan termin, do tada uobičajen samo u sferi mejnfrejm računara, postaje ključna reč u svakodnevnom govoru hakeru.**

Termin akcelerator lako se prevodi — to je ubrzivač. Međutim, on ništa ne govori o tome šta se ubrzava i kakvo ubrzanje se njime postiže. Pre no što ovaj zvučni i gotovo mistični termin svedemo na pravu meru, objasnimo šta se sve u računaru može ubrzati.

## Šta ubrzavamo?

Osnovni vid operacija koje CPU izvršava su aritmetičko-logičke operacije, pristup memoriji, pristup ulazno-izlaznim (I/0) uređajima i u okviru njih komunikacija sa spoljašnjom memorijom. Ako izostavimo manje važne I/0 uređaje, možemo izdvojiti tri osnovna pristupa ubrzanju rada: ubrzanje procesora, ubrzanje radne memorije i ubrzanje pristupa spoljnoj memoriji. Počnimo od perifernih problema.

## Ubrzavanje I/0 operacija

Problem ubrzanja pristupa disk-jedinicama i hard diskovima nije jednostavan. Obzirom da brzina I/0 operacija zavisi od vremena pristupa disku (to jest od brzine rotacije, brzine lateralnog kretanja glava i drugog) a ne od toga kako brzo procesor piše i čita sa portova (razlika je nekoliko redova veličine) jasno je šta treba ubrzati. Rešenje može biti implementirano kroz hardverske, odnosno softverske ubrzivače, pri čemu oba varijante rade na principu keširanja.

Na tržištu postoji relativno mali broj hardverskih ubrzivača i svi se odlikuju visokom cenom i ne baš zadovoljavajućom kompatibilnošću. Vredi spomenuti karticu **Awesome I/0 Card** firme **CSSL, Inc** koja, koristeći specificirani format zapisa i naročite tehnike kompresije, ubrzava hard disk i povećava mu kapacitet uz opcionu korekciju greške. Mada se vreme pristupa može smanjiti 5—12 puta, cena od 650 dolara za osnovnu verziju plus 250 dolara za disk keširajući softver i softver za korekciju grešaka čine ga slabo prihvatljivim izborom. Nešto povoljniji odnos cena/karakteristike ima **Konan Tentime Disk Controller** gde postoje verzije za ubrzanje disk jedinice, hard diska ili obe zajedno. Nešto razumnija cena od 300 dolara za osnovnu verziju, međutim, ide na uštrb kompatibilnosti sa izvesnom količinom hardvera i softvera. Ček i da nije tako, još uvek se više isplati kupiti brzi hard disk nego pomenutim proizvodima ubrzavati postojeći.

Obzirom da hardverski pristup problemu ne daje elegantna i isplativa rešenja, alternativa je potražena i pronađena u disk-keširing softveru. Takav softver omogućava ubrzanje pristupa diskovima 3—10 puta, ali je različitog tipa i uz to veliki deo memorije rezerviše za disk keš. Ove mane su kod velikog broja primena uglavnom zanemarljive, dok je ubrzanje u nekim slučajevima zaista enormno. Među najkvalitetnijim proizvodima ističe se *Disc Cache/Speed Write* (ovaj drugi vrši „background" upis na disk) firme **Microcosm** koji omogućava upotrebu do 4 MB RAM-a kao disk keša po ceni od 100 funti za oba programa zajedno. Tu je i nešto brži *Fast Forward* firme **Mark Williams Company** kao i *Lightning* firme **PCSG** koji se odlikuje visokim stepenom kompatibilnosti a može se dobiti besplatno uz kupovinu neke od proizvođačevih akceleratorskih kartica.



## Ubrzavanje memorije . .

Najprimitivniji način ubrzavanja memorije je zamena postojećih RAM čipova nekim bržim. Ovakav zahvat se retko kad (čitaj nikad) preporučuje, jer je odnos cena/performanse po pravilu katastrofalan. Uz to, velika je mogućnost da se zamenom čipova neće ništa postići pošto razlika između deklarisanih vremena pristupa nije tako velika kao što se obično misli. Recimo, čipovi od 150 ns ne moraju da budu ništa sporiji od onih sa oznakom 120 ns. Razlika može biti samo u deklaraciji proizvođača, pošto ovi prvi nisu prošli izvesne fabričke testove, čime su automatski svrstani u nižu kategoriju. Na složenije metode ubrzavanja memorije osvrnućemo se nešto kasnije u okviru pregleda metoda ubrzavanja procesora i akceleratorskih kartica.

## . . . mikroprocesora . .

Pređimo stoga na ubrzavanje onog koji je odgovoran za obavljanje većine operacija sa memorijom. I ovde se mogu izdvojiti trivijalni i sofisticirani metodi. Mada su teme članka ovi drugi, za njihovo razumevanje neophodno je pozabaviti se fundamentima procesa ubrzanja. Najjednostavniji način je opet metod zamene. Intelov par 8088/86 može se zameniti sa „pin to-pin" kompatibilnim NEC ovim mikroprocesorima V20 i V30. Zahvaljujući usavršenoj arhitekturi i izvršavanju nekih instrukcija u manjem broju ciklusa, ovi procesori u realnim aplikacijama postižu ubrzanja od 10—18% (pri istom kloku, naravno). Obzirom na zanemarljivost postignutog ubrzanja, ovakav postupak se isplati samo ako je srce vašeg ljubimca već prestalo da kuca. Cene ovakve zamene u Americi iznose 16—35 dolara, dok je cena u Evropi po tužnom pravilu gotovo dvostruko veća. Ovo, na žalost, važi i za ostale cene navedene u tekstu.

Nešto složeniji metod je ubrzanje takta zamenom postojećeg čipa takt-generatorom koji generiše višu frekvenciju. Rekavši „složeniji" mislimo na brojne nevolje koje ovaj postupak može da prouzrokuje. Ubrzavanje takta moguće je samo do određene granice, jer memorija neće moći večno da ide ukorak sa procesorom. Nije zanemarljiv ni problem povećane disipacije koja ubrzano skraćuje vek procesora čak i pri minimalnom ubrzanju takta. Treće, ukoliko RAM čipovi nekim čudom i uspeju da slede procesor, niko ne garantuje da će, na primer, vaš DMA čip biti raspoložen da uredi isto to, pa vam sledi njegova zamena. Isto može da važi i za još poneki kontroler. Treba pomenuti i eventualne probleme sa malim brojem programa čije zaštite reaguju na promenu takta (stare verzije *Lotusa* i *Framework*-a) mada je to danas prava retkost. Troškovi ovakve zamene mogu iznositi i preko 40 dolara.

Oni kojima se dotični postupak čini suviše složenim mogu izabrati jedan od više modela kartica koje na sebi nose novi generator, pa čak i kombinaciju generator/DMA. Ovakve kartice se obično nameštaju na podnožje postojećeg generatora i kablom povezuju sa DMA kontrolerom. Cena tog luksuza je bar tri puta veća od prethodnog postupka, što uz minimalna ubrzanja koja se postižu ove kartice čini prilično nezanimljivim. U Americi je prilično popularna kartica **Microspeed Fast 88**, a njoj je slična kartica **PC Bandit** koju smo opisali u „Računarima 31". Ugradnjom ovakve kartice vašeg teško stečenog ljubimca dovodite u opasnost, naročito ako je lajvanske proizvodnje. U prevodu, 8088 i prateći čipovi jedva su izdržali „burn-in" testove na 5 MHz (ako su im uopšte bili podvrgnuti) pa će instaliranje ovakve kartice pospešiti skori odlazak procesora u „večna lovišta" s tim što će procesor najverovatnije povući sa sobom i gomilu čipova. Posle popravka računara utvrdićete da biste jeftinije prošli da ste kupili **Turbo-XT** ploču a staru bacili u đubre.

## . . . i koprocesora

Sledeći način ubrzavanja spada u trivijalne metode a prihvatljiv je za numerički orijentisane primene. Intelova familija 80×87 kao i **Weitek-ov 1167** (primenljiv za sada samo na nekim 80386 računarima i **Quad 386 XT** akceleratoru) drastično ubrzavaju rad sa transcedentnim funkcijama. Uz veliku cenu (naročito za 10—20 MHz) koja premašuje cene mnogih akceleratorskih kartica, najveća mana im je to što im je primena ograničena na programe koji su pravljeni tako da iskoriste njihovo prisustvo. Nabavka koprocesora će se još jednoj grupi korisnika isplatiti i to veoma. Naime CAD programi često izvršavaju složene trigonometrijske funkcije te će koprocesor znatno ubrzati rad sa crtežom (CUT PASTE ROTATE). Kako su „Računari" o koprocesorima već podosta pisali, nećemo ih detaljnije razmatrati.

Onima koji nisu zadovoljni brzinama koje postižu koprocesori, američki proizvođači nude veći broj ubrzivača za koprocesore (izgleda da nijedan čip u računaru nije pošteđen maltretiranja). Najjednostavniji je **Co-Clock**, proizvod firme **SPI**. Zasnovan je na činjenici da kod velike većine 80286 računara koprocesor radi na frekvenciji koja je 2/3 niža od deklarisane za 80286. **Co-Clock** je pločica koja se smešta na podnožje koprocesora na koju se, zatim, postavlja 80287, a sadrži i hibridno kolo koje obezbeđuje rad koprocesora na dozvoljenoj frekvenciji. Pored toga, proizvođač tvrdi da je mnogo sigurnije koristiti koprocesor pomoću **Co-Clock**-a nego ostaviti računaru da ga snabdeva varijabilnom frekvencijom koja vodi neprimetnim kalkulativnim greškama, mada takve teze ostaju samo u domenu teorije. Cena mu je oko 70 dolara, i pitanje je koliko je nabavka isplativa.

Sličan pristup ima i **Microway**-ev koprocesorski akcelerator **287 Turbo** koji se po ceni od 450—550 dolara prodaje sa ugrađenim 80287 koji radi na 10 ili 12 MHz. Drugi interesantan proizvod ove firme je **287 Turbo Plus** (150 dolara bez koprocesora). Sastoji se od dve ploče pri čemu se prva montira na mesto koprocesora, a druga sa zadnje strane računara. Na ovoj se nalazi potenciometar pomoću kojeg se kontinualno može povećavati frekvencija glavnog procesora i koprocesora do maksimalnih 11,5 MHz. Frekvenciju povećavate dok vas ne spopadnu nevolje a zatim je malo smanjite. Ovo je idealan način da upropastite računar iako ćete uz opreznu upotrebu osetno povećati brzinu rada. Kao i kod prethodnih kartica, i ovde je odnos cena/kvalitet prilično slab.

## Podela akceleratora

Nakon pregleda osnovnih metoda ubrzavanja, vreme je da izvršimo podelu akceleratora. Po nameni delimo ih na disk akceleratore, (ko)procesorske akceleratore i akceleratore posebne namene. Prve smo već podelili na hardverske i softverske. U procesorske akceleratore se ubrajaju jednostavni ubrzivači takta, brzi kompatibilni procesori, aritmetički koprocesori i akceleratorske kartice različitih konstrukcija. Akceleratora posebne namene ima veliki broj te ih je teško svrstati u kategorije. Grubo rečeno, to su kartice koje omogućavaju primenu PC-ja u specifičnim aplikacijama. Na primer, to su profesionalni razvojni sistemi sa C, fortran, lisp ili forth radnom okolinom, zatim sistemi za akviziciju podataka za FFT i spektralnu analizu i slično. Ovde spadaju i sistemi sposobni da velikom brzinom obrađuju kompleksne naučno-tehničke probleme između ostalih, (transpjuterske kartice i vektorski koprocesori), kao i mnogi grafički kontroleri koji su u osnovi akceleratori ali i mnogo više od toga, pa zaslužuju poseban prikaz. U akceleratore možemo svrstati i kartice koje uvode moćnije operativne sisteme kao što su UNIX, XENIX, zatim industrijske sisteme za rad u realnom vremenu kao što su RTOS i OS-9, te brojne multikorisničke i multitasking sisteme, a kojima braća 80×8x nisu niti će tako skoro dorasti. Sve ove kartice zbog svoje raznolikosti i ograničene primene ne mogu biti tema ovog članka.

Tako smo izbor suzili na „univerzalne" akceleratore, koji bi se tako mogli nazvati zbog toga što zadržavaju potpunu MS-DOS kompatibilnost, uz ubrzanje većine pomenutih operacija, omogućavajući da posustali PC-XT dostigne, ako ništa drugo, a ono bar brzinu rada AT računara, mada je u izvesnim slučajevima daleko prestiže.

## O ovome ni reči

Oni koji bolje poznaju problematiku akceleratora mogu zameriti što se u pregledu nisu našle neke široko rasprostranjene ili barem nadaleko reklamirane kartice. Razlog je u tome što akceleratore danas delimo na tri generacije. Prva se pojavila pre pet godina i koristila je 8086 i 80188/6 procesore za ubrzanje prvih sistemskih ploča, koje su uglavnom bile konstruisane oko 8088. Druga generacija uvela je 80286 a pojavila se vrlo brzo po izlasku AT-a na tržište. Većina tih akceleratora ima napredniji pristup metodama ubrzavanja, a najčešće koristi tehnike memorijskog i disk kešinga. Treća generacija sadrži 80386 i mnogo rafiniraniju konstrukciju, koju omogućava prisustvo savremenih procesora.

U kategoriju veterana, odavno prevaziđenih na polju konstrukcije i brzina, spadaju na primer **Classic Speedpack, Orchid Turbo EGA** i **Univation Dream Board**. Iako izrađeni u vreme kada su hit bili 80188/6 i danas su ostali primer vrhunskog dizajna, ali i astronomskih cena. Krasio ih je visok stepen kompatibilnosti sa različitim modelima računara, a mnogim korisnicima još uvek dobro služe bez obzira što ih danas po svim parametrima prevazilaze i četiri puta jeftinije kartice.

Nadaleko razvikani **PC Elevator 286** pokazao je potpunu nekompatibilnost sa *DBase III*, *Flight Simulator*-om i još gomilom softvera, a uz to je prosto „alergičan" na prisustvo nekih grafičkih kartica. Slično važi i za modele **PMI 80286** i **PMI Fast Card** firme **Peripheral Marketing Inc**. Ove kartice uz niaku cenu sadrže i neka vrlo lucidna konstruktivna rešenja, ali je zato spisak računara na kojima ne mogu da prorade toliko da ne bi stao na ovu stranu „Računara".

**Microway Inc.**, firma koja u svom proizvodnom programu ima čitavu paletu kvalitetnih akceleratora, najnovijim modelom **Supercache 286** je, izgleda, stavila sebi prst u oko. Prema stranim izvorima, ovaj akcelerator ne može da se instalira u čitavom nizu sistema. Egzotičnu gamu čini i dosta reklamirani, veoma brzi ali isto toliko i nekompatibilni **Attak 286 model II/24 MHz** firme **A/T**, koji izgleda zahteva još dorade, mada je nasledio ionako mušičavu karticu **Attak 286-1A**.

Po nekim kriterijumima, u univerzalne akceleratore mogli bi da se svrstaju i oni iz proizvodnog programa firme **Processing Data Management**, proizvođača koji se specijalizovao za kartice prilagođene operativnim sistemima za rad u realnom vremenu. Pre izvesnog vremena ova firma je na tržište izbacila i nekoliko akceleratora za ugradnju u PC, ali istovremeno deklarišući da oni nisu 100% kompatibilni sa AT-om. Većina radi na frekvencijama od 20—32 MHz. No, kako o njima još nema relevantnih podataka, uz neke proizvođače i modele koji na ovaj ili onaj način nisu zaslužili da se nađu u našem pregledu, moraće da budu izostavljeni. Posebnu grupu čine za sada malobrojni XT akceleratori bazirani na 80386 procesoru. O njima i mnogim drugim modelima više reči u sledećem broju „Računara".

*Nenad Vereš*

## Najbrži na svetu

Već mnogo puta se desilo da inovatori svojim pronalascima nadmaše i najsmelija futuristička predviđanja, bar kada je u pitanju razvoj računarske tehnike. Ono što je nedavno uradila firma **Sky Computers Inc**, zaista je malo ko mogao da očekuje. Niko nije predviđao da bi se u tako kratkoj istoriji personalnih računara mogle dostići brzine iz „near-supercomputer" klase. Pomenuta firma od februara ove godine izbacuje na tržište VORTEX seriju 64-bitnih vektorskih akceleratora i svakog meseca je dopunjuje novim proizvodima prilagođenim za rad na različitim personalnim računarima i radnim stanicama. Zasad su raspoloživi sistemi koji se mogu priključiti na PC AT kompatibilne računare, kao i računare opremljene MULTIBUS II, Nu Bus i VME interfejsima.

Brzine koje VORTEX postiže su zaista impresivne. Firma deklariše maksimum od 20 MFLOPS-a 32-bitnih operacija u normalnoj odnosno 10 MFLOPS-a 64-bitnih operacija u dvostrukoj tačnosti. Akcelerator se sastoji od pet glavnih modula (ili blokova) koji su smešteni na tri kartice. Ove kartice zauzimaju samo jedan slot, jer se dve manje kartice koje sadrže memoriju i predvektorizator vezuju direktno na glavnu karticu sa brzinom prenosa od 80 MB/s. Blokove sačinjavaju sistemska magistrala, radna memorija, kontrolni procesor, memorija za podatke i vektorska aritmetička jedinica.

Vektorska aritmetička jedinica izrađena je u „bit-slice" tehnici, a sadrži **Analog Devices** čipove AD 3210 i AD 3220 (veoma brze jedinice za sabiranje i množenje u pokretnom zarezu), WAFER SCALE-ovu 32-bitnu registarsku datoteku i aritmetičko logičku jedinicu WS 5903, kao i AD 1401 sekvencer. Ostali čipovi su kastomizovani VLSI, PAL i GAL tipa i čine protočne registre i bafere. Memorija za podatke u sadašnjoj verziji je 1 MB i izgrađena je od 256 Kbitnih DRAM i SRAM čipova, uz mogućnost proširenja do 8 MB pomoću 1 Mbitnih čipova. Na PC AT-u ove memorije se uklapa u standardni adresni prostor računara. Brzi SRAM čipovi opslužuju vektorske aritmetičke jedinice, dok se sporiji DRAM čipovi (njih ukupno 8 KB) uglavnom koriste za manje zahtevne zadatke, kao što je recimo transfer podataka preko sistemske magistrale. Ovakva koncepcija je primenjena da bi se cene svele u što realnije okvire. Tako se sada kompletan sistem sa osnovnim softverom može dobiti za oko 9000 dolara a po performansama dostiže i neka super mini računare za koje je potrebno platiti i do milion dolara. Da bi se omogućilo proširenje postojećeg seta instrukcija, kao i emulacija setova instrukcija drugih procesora, primenjena je tehnika mikrokoda. Njegovo korišćenje omogućava kontrolni procesor. Sve je urađeno u CMOS tehnici da bi se smanjila ionako velika potrošnja. Da bi se 198(!) čipova smestilo na relativno mali prostor, ekstenzivno je korišćena SMD tehnologija.

Najlepša mogućnost kojom su konstruktori opremili VORTEX a je sposobnost automatske vektorizacije. Za to je zadužen VEX vektorizujući pretprocesor, sposoban da pregleda standardne fortran 77 i C programe (za sada samo njih) i transformiše ih u vektorski izvorni kod koji je moguće prevesti običnim kompajlerom. Ono što se ne može vektorizovati, VORTEX će obraditi na svojim skalarnim jedinicama. Planiraju se nove verzije VEX a sa mnogo većim stepenom vektorizacije, kao i pretprocesori za druge jezika.

Korisnik kome brzina jednog VORTEX a nije dovoljna može da ih doda još (u idealnom slučaju kod PC AT klonova 4—5) koja rade simultano i teoretski mogu da dostignu brzinu od gotovo 40 MFLOPS-a 64-bitnih operacija. A to je, uz odgovarajuću cenu, više nego dovoljno da proizvođači neuporedivo skupljih „near — supercomputer-a" pozelene od zavisti.

*Dejan Predić*

Komercijalni softver

Programski jezici/FORT

# Fort bez issimo

**Programski jezici nastaju na različite načine. Kobol i „adu" su napravili timovi specijalista po nalogu Ministarstva odbrane SAD. FORTRAN je nastao da bi sa olakšalo računanje; PL/I kao „interni" jezik korporacija IBM; a algol (1960 godina) da bi stručnjaci za računare sami sebi objasnili šta je to dobar programski jezik. Paskal ja stvoren da bi se dokazalo da struktuirano programiranje ima smisla u praksi, modula-2 da bi se mogao napisati operativni sistem za računar *Lilith*, C da bi se UNIX lakša prenosio na nove mašine, a smoltok da bi deca mogla da programiraju i koriste računare . . . Programski jezici su, dakle, stvarani da bi se rešili neki osnovni problemi korišcenja računara. Jedino je fort nastao da bi se uz pomoć računara rešavali neki drugi, praktični problemi. Razlika u pristupu je fundamentalna, pa fort ostaje najneshvaceniji programski jezik današnjica.**

Fort je „nešto drugo" u odnosu na sve ostale programske jezike. Teoretski lako ga je koristiti jer je interaktivan brz je zato što je prevodilac ali ne i tako brz poput čistog mašinskog jezika jer je ipak i interpreter. Ali takođe je tačno da je teži za učenje u odnosu na druge jezike. Osnova forta su takozvane reči koje odgovaraju funkcijama potprogramima i procedurama iz C a fortrana i paskala. Skoro svaka upotrebljiva reč (bilo da se već nalazi u jeziku ili da je mi sami pravimo) poziva druge reči i upravo zbog te otvorenosti broj reci koje treba odjednom držati u glavi je veliki. Reči se nalaze u rečniku (*dictionary*) s tim da unutar rečnika postoje leksikoni (*vocabulary*) — podgrupe reči od posebnog interesa za programera. Na primer sve reči u vezi sa asemblerom (obavezan sastavni deo svakog forta) nalaze se u leksikonu ASSEMBLER. Program nastaje kombinacijom novih programerovih reči i reči koje se već nalaze negde u rečniku sve to povezano standardnim upravljačkim naredbama poput IF ELSE ENDIF ili DO LOOP

## Nove reči

Nove reči obično nastaju i isprobavaju se za tastaturom dakle interaktivno a podrazumeva se da se mogu učitavati i iz datoteka. Mogućnost interaktivnog testiranja novih reči je osnova za povećanje programerske produktivnosti, jer to znači da se u program unose samo istestirani potprogrami. U pravim rukama, fort može spektakularno povećati brzinu pisanja programa ali loš programer ostaje loš i sa fortom i bez njega.

U odnosu na „normalne" programske jezike, prvi koraci u fortu su čudni. Iako postoje i konstante i varijable pravi fortprogramer ih retko koristi zato što ima potpunu kontrolu nad stekom. Mala većina programera toga nije svesna, interni stek postoji bukvalno u svakom programskom jeziku i koristi se za izračunavanje aritmetičkih izraza. Zato se recimo u paskalu može pisati na način koji je sasvim sličan onom u školskoj matematici, na primer

A = 2 + 5

Evo šta se tom prilikom događa. Brojevi 2 i 5 dovode se na vrh steka tu se sabiraju sabiranje ih uništava a na vrhu steka ostaje rezultat kojeg prevodilac (zbog naredbe pridodeljivanja) smešta na adresu varijable A. Ni nad kojim delom tog procesa programer nema nikakvu kontrolu u fortranu paskalu i ostalim jezicima tog tipa. Međutim, u fortu programer „ručno" izvodi ceo proces, tako što sam postavi oba broja na stek sam ih sabere i sam odlučuje gde će rezultat biti smešten (da li da ostane na vrhu steka da bi učestvovao u nekoj sledećoj operaciji ili da bude presikan u memoriju). Tehnički fort programer bi to izveo veoma jednostavno, jer se prostim navođenjem broja u programu on automatski smešta na vrh steka i potiskuje sve ostale brojeve za po jedno mesto naniže. Dodatnu olakšicu pruža i činjenica da se operacija sabiranja (kao i većina operacija u fortu, uostalom) odnosi upravo na brojeve na vrhu steka. Sve zajedno sabiranje se u fortu piše ovako

2 3 +




Na žalost, takva notacija upravo je suprotna uobičajenom načinu pisanja u matematici i životu i upravo je to otežavajuća okolnost za početnike u fortu! Fort je zato — sa izuzetkom asemblera — jezik najnižeg nivoa u poređenju sa svim ostalim popularnim jezicima. Zato je s pravom nazvan i asemblerom najvišeg nivoa.

U fortu programer ima računar „kao na dlanu", sve je dozvoljeno — samo treba razmišljati u terminima hardverskih elemenata adresa stek sabiranje i slično. Memorija je sasvim nestruktuirana i nezaštićena baš kao u pravom mašinskom jeziku. Ali fort je i „nešto više". On je zapravo meta-jezik za definisanje programskih jezika. To znači da programer može da definiše kompletne nove programske jezike. I zaista mnogi bejzici, pa čak i jedan paskal, napisani su upravo na fortu!

U metodološkom pogledu, fort je dobro prilagođen metodi „odozgo nagore" (bottom-up) za razvoj programa. Naravno, ništa ne može zameniti generalno osmišljavanje programa (metoda „odozgo-nadole"), ali tehnikom „odozdo nagore" lako se ispituju algoritmi interfejsi i brzo se stiže do radne verzije programa. Time se mogu otkriti greške u specifikaciji programa a krajnji korisnici mogu odmah da kritikuju ili hvale program što sve zajedno može da ubrza stvaranje programa.

## Osnovne naredbe

Zbog steka i za sve ostale naredbe prvo se navode argumenti pa tek onda operator. Tako je sintaksa naredbe IF sledeća

**zastavica IF .. ENDIF**

Zastavica je broj na vrhu steka. Ako nije nula, izvršavaće se reči između IF i ENDIF, na primer

**: TEST IF ." vrh steka nije nula " ENDIF,**
**1 TEST vrh steka nije nula ok**
**0 TEST ok**

Gornji primer ilustruje još neke sintaksne detalje. Nova reč se definiše navođenjem naredbi između : i ;. Prva sintaksna jedinica (**TEST** u ovom primeru) postaje nova reč a ostatak je ono što reč treba da radi. Naredba ." štampa string, zaključno sa sledećim dvostrukim navodnikom. Telo naredbe **TEST** je IF, pa prilikom korišćenja nove reči **TEST** treba se ponašati kao da je u pitanju IF -- a znamo da IF očekuje broj na vrhu steka. Zato je u drugom redu gornjeg primera ispred TEST došao broj 1. Ostatak

### Tehnički detalji

Za ovaj prikaz firma MicroProcessor Engineering Ltd poslala je *Work–FORTH* rev 1.56. Na svega dve diskete od 5.25 inča nalaze se četiri skladno ukomponovana programa. Osnovni je, naravno, sam *Work FORTH* sa priručnikom od 120 strana a uz njega dobro dođu i *Floating Point Package* (27 strana) *View-Trace* (23 strane) i *Documentation ToolKit* (10 strana). Cena za sva četiri paketa je 110 a samo za Work-FORTH — 55 funti. Adresa je MicroProcessor Engineering Ltd 133, Hill Lane, Shirley Southampton SO1 5AF England Great Britain tel 9944 703 780084

Priručnici su pisani spartanski po ugledu na način pisanja programa u samom fortu. Iako glavni priručnik sadrži poglavlje „Brzi uvod u FORTH", ono je nedovoljno da potpunom početniku omogući rad sa ovim jezikom. Ostatak dokumentacije bavi se programskim editorom asemblerom, datotekama a tu je i neizbežan rečnik svih raspoloživih reči. Uputstva — iako kratka — pisana su jasno i pregledno.

Firma MPE održava i prodaje ovaj isti fort i na računarima pod operativnim sistemom CP/M kao i za računare „amstrad PCW". Razlika je u asembleru i u interfejsu prema operativnom sistemu a sve ostalo je isto. Verovatno zbog toga *Work FORTH* uopšte ne radi sa grafikom PC računara! Valjda se — u skladu sa tradicijom fort-programiranja — očekuje da programer sam napiše sve što mu proizvođač nije isporučio!?

*Work FORTH* potpada pod takozvani FIG-FORTH standard. To je dobro jer ipak je to najrasprostranjenija i najpodržanija verzija forta. (Ostali standardi su FORTH79 i FORTH83. Razlike postoje ali nisu nepremostive)

## Instalacija

Instalacija ovog programa je veoma laka, ukoliko poseban program za to nije isporučen. Svodi se na otvaranje novog imenika na tvrdom disku i kopiranje obeju disketa običnom naredbom COPY A * * iz DOS-a. Program se poziva navođenjem imena WRKFORTH.

Iz istorijskih razloga, svi fortovi se isporučuju sa linijskim editorom, koji radi nad skupom od 16 linija dužine 64 znaka. *Work-FORTH* ima ovaj editor, ali se preporučuje korišćenje posebnog editora firme MPI, po imenu *FRED*. Njega treba instalirati za svaki terminal posebno, ali ako je *Work-FORTH* kupljen za tačno određenu klasu računara, onda će to dodatno predefinisanje izvršiti sam proizvođač.

reda (posle 1 TEST) je izvršena naredba TEST za slučaj da je logička vrednost na vrhu steka bila TRUE. Treći red pokazuje da je nula logički „negativna" u fortu.

Slična naredba je i IF ... ELSE ... ENDIF s tim da se između ELSE i ENDIF nalaze reči koje će se izvršiti ako je nula na vrhu steka.

*Work FORTH* je nešto bogatiji petljama od uobičajenih fortova. Za početak, tu je konstrukcija DO ... LOOP sa sledećom sintaksom

**granica indeks DO <telo petlje> LOOP**

Slična je i naredba DO ... +LOOP, njoj u bejziku odgovara

**FOR X=1 TO 10 BY ... <telo petlje> NEXT**

Petlja BEGIN ... AGAIN završava se samo ako se dogodi neka izuzetna situacija (greška), ili ako se izvrše reči ABORT i QUIT.

KEY ispituje tastaturu i vraća redni broj pritisnutog tastera. Njega treba uporediti sa 13, jer je to redni broj tastera Enter. Posle KEY dolazi naredba DUP, koja dovodi kopiju vrha steka na vrh steka tako da su prva dva broja na vrhu steka uvek identična posle ove reči. Poređenje sa 13 briše jednu od te dve vrednosti, pa na steku ostaje broj pritisnutog tastera. Ako je u pitanju baš Enter, izvršava se reč ABORT što kasnije omogućava izlazak iz petlje. Konačno, reč EMIT prihvata broj sa steka i prikazuje odgovarajući znak na ekranu.

Slična petlja je BEGIN ... UNTIL. One se obavezno izvršava bar jednom.

Konačno, tu je i klasična petlja BEGIN ... WHILE ... REPEAT. Petlja počinje sa BEGIN i izvršavaju se reči sve do WHILE. Ako na vrhu steka nije nula, izvršiće se i reči između WHILE i REPEAT. Petlja zatim ponovo počinje od BEGIN. Ove naredbe je vrlo moćne, i nije sasvim identična sličnim naredbama u C-u ili paskalu.

**Opšti oblik naredbe CASE je:**

**CASE**
**vrednost1 OF ... ENDOF**
**vrednost2 OF ... ENDOF**
**. .**
**<hvatanje greške>**
**ENDCASE**

Uzgred, gornja CASE naredba je pobedila na konkursu časopisa *FORTH Dimensions* za najbolju CASE naredbu. Od mnogih varijacija na tu temu, gornja naredba je najsigurnija, najlogičnija i najlakša za upotrebu.

U fortu je string neki niz znakova. Standardni fort ne nudi mnogo reči za rad sa stringovima, pa ni *Work-FORTH* tu nije bolji. Programeru je ostavljeno na volju da programira stringove sam ili da u tu svrhu koristi neku pogodnu biblioteku reči.

Rad sa datotekama je rešen bolje nego u drugim fortovima, ali ipak ne tako dobro kao u drugim programskim jezicima. Datotekama se pristupa kroz takozvane upravljačke blokove datoteka (File Control Block, skraćeno FCB). Na primer,

**FCB FRED**

otvara jedan takav blok i rezerviše prostor za njega. Sva daljnja obraćanja datoteci FRED koristiće taj bafer u memoriji kao i standardna osobina operativnog sistema. Zastupljene su sve uobičajene naredbe: otvaranje i zatvaranje datoteka, preimenovanje, direktan i sekvencijalni pristup podacima, pozivanja interaptova 021H i 024H, listanje sadržaja imenika i slično. Naredba SAVE snima kompletan *Work-FORTH* zajedno sa svim novim rečima na disk, tako da u sledećem učitavanju programer može da nastavi tamo gde je stao. Kada je program završen, moguće ga je prevesti u samostalan program, sa nastavkom EXE. Osim toga, MPE podržava petnaestak unakrsnih prevodilaca za *Work-FORTH*, što znači da se program može pisati na jednom računaru, a da ga je moguće izvršavati na procesorima Z80, 8085, 68000, 6805, 6801 – 6083.

## Aritmetika pokretnog zareza

Fort je lako proširivati u svim mogućim pravcima. Iz istorijskih razloga, fort se po pravilu isporučuje bez reči za računanje sa brojevima u pokretnom zarezu. To je naravno veliki hendikep i jedan od važnijih razloga što fort nije doživeo veću popularnost. MPE zato nudi i *Floating Point Package*, kroz koji se dodaju navedene aritmetičke operacije i dodatne naredbe za stringove. Odgovarajuće reči se nalaze u datoteci EPLIST.SCR, odatle se učitavaju kao program u fortu. Sve su date u izvornom obliku, osim dve (za kružno premeštanje sadržaja registara).

Naredbe za štampanje, sabiranje, oduzimanje,

## FRED

Programeri komuniciraju sa računarom na dva nivoa: konceptualno (šta i kako treba nešto napisati) i fizički (tastatura, ekran, editor). Bitno je da editor ne sputava kreativan rad, odnosno da je što bolje prilagođen filozofiji jezika s jedne strane i navikama korisnika sa druge strane. *FRED* bi se najkraće mogao opisati kao „moderna izvedba zastarele koncepcije". Preciznije, zadržan je ekran dimenzije 64×16, ali su naredbe interaktivne. Korisnik može sam da predefiniše sve opcije u *FRED*-u, što je u priručniku i objašnjeno. Većina korisnika će to verovatno i poželeti da uradi jer naredbe *FRED*-a nisu po *Wordstar* standardu.

Dve osnovne naredbe su USING i EDIT. Prvo naredbom USING naznačavamo ime programa, a zatim ga učitavamo sa EDIT. Tako će naredbe

USING RND-GEN
EDIT

učitati ekrane programa RND-GEN i prikazati ih na ekranu. Kada je već jednom editor učitan u njega se može (ponovo) ući naredbom poput

25 FRED

što bi aktiviralo editor i prikazalo 25-ti ekran.

Sa desne strane programskog teksta nalaze se obaveštenja o verziji *FRED*-a, kao i da li se unos teksta odvija u režimu umetanja ili ne. Do ekrana sa opisom tastera (HELP) dolazi se pritiskom na Esc H. Sve uobičajene editorske naredbe su tu, ali mnemonički raspoređene po tastaturi. Tako Ctrl-L pomera kursor ulevo, Ctrl-R udesno, Ctrl-U na gore, Ctrl D na dole, Ctrl F pomera za reč udesno, a Ctrl-B pomera za reč ulevo i tako dalje. Ostale naredbe se koriste slično, uz dodatak tastera Esc kao početnog u nizu. Korisnik može direktno da pređe na neki drugi ekran (mora biti naveden njegov redni broj), može da kopira ceo ekran u neki drugi i slično.

*FRED* ima još i line stack (linijski stek) i barrel (rotacioni stek). Oba služe za prikupljanje linija koje treba preneti u druge ekrane. Rotacioni stek je identičan običnom steku, osim što posle poslednje linije sažemo ponovo do prve i tako u krug. Date su i osnovne naredbe za rad sa rotacionim stekom: umetanje linije iz ekrana, brisanje linije, vraćanje linije iz rotacionog steka u (neki drugi ili isti) ekran, brisanje celog steka.

*FRED* je kompletno napisan u *Work-FORTH*-u, i priložen je u obliku izvornog programa. Zainteresovani korisnik može tako da ga promeni, iznova prevede i tako dobije sopstvenu verziju interaktivnih naredbi.

Ocena ovog editora mora biti kontroverzna. Ako ste već imali nekih iskustava sa fortom na drugim računarima, *FRED* će vam izgledati kao najbolji editor na svetu. Programerima koji su navikli na blagodeti Borlandovih integrisanih editora, *FRED* izgleda arhaično.

množenje i deljenje pišu se ovako

F., F+, F–, F*, F/

Zastupljene su sledeće elementarne funkcije: FSQR (kvadratni koren), FSIN (sinus ugla u stepenima), FASIN (inverzni sinus), FCOS (kosinus), FACOS (inverzni sinus), FTAN (tangens), FATAN (inverzni tangens), FLN (prirodan logaritam), FLAN (prirodan antilogaritam), FLOG (logaritam za osnovu 10) i FALOG (antilogaritam za osnovu 10).

Princip je jasan: za ovu vrstu aritmetike postoje sve standardne reči, jedino što počinju sa F.

Raspon brojeva u pokretnom zarezu je od EXP 99 do EXP+99.

## Dokumentovanje programa

Programska dokumentacija je najzoslavniji deo samog programa, ipak najčešće se piše tek pošto je sam program završen. Zato MPE isporučuje *Documentation Toolkit*, program koji automatski stvara neke delove dokumentacije. To je fortu olakšano činjenicom da su sve reči ionako u rečniku. *Documentation Toolkit* sortira reči, a zatim formira njihov unakrsni indeks, to jest za svaku reč pravi spisak drugih reči u kojima se koristi. To je značajno prilikom menjanja već postojećih reči. I ovaj program je dat u izvornom obliku, a interesantan je za čitanje zbog primenjenih programskih rešenja (dvostruko povezane liste, sortiranje i slično).

## Dibager *View Trace*

Za skoro sve moderne jezike postoji i dibager. Uz *Work FORTH* može se koristiti *View-Trace*. Početnicima će to omogućiti da nauče kako fort radi, a iskusni korisnici će ga koristiti kao moćan dibager. *View-Trace* bi trebalo da prikazuje četiri različite informacije istovremeno:

— Izvorne naredbe reči čije se izvršenje prati
— Do deset „najviših" elemenata steka
— Nekoliko elemenata povratnog steka
— Poslednje izvršene reč u rečniku, što je bitno kada se uvode nove definišuće reči

Naredbe čije se dejstvo analizira trebalo bi da budu osvetljene inverzno. To jednostavno nije radilo u testiranoj verziji *Work FORTH*-a, pa se bitan vizuelan element ovog dibagera gubi. Inače, prikaz na ekranu veoma je sličan *FRED*-u (editoru za *Work FORTH*).

Program se mora zasebno prevesti sa priključenim *View-Trace*-om, i tada se izvršavanja u mnogome usporava. Ako se ne navede drukčije, *View-Trace* će izvršavati program korak po korak. Međutim, to se može i ubrzati i (kasnije) usporiti, po potrebi.

*View Trace* omogućava programeru da interaktivno menja radnu sredinu, na primer vrednosti na stekovima, da izvrši neku drugu naredbu (najčešće BINARY, OCTAL, DECIMAL ili HEX) i slično.

## Sve u svemu

Ocena *Work FORTH*-a mora biti pomalo kontradiktorna. Radi se o veoma standardnom fortu, toliko standardnom da kod programera može izazvati osećaj da radi na „starom dobrom" „sinkleru" ili „šarpu". To znači i da se na ovu verziju forta mogu primeniti sve mnogobrojne knjige o fortu, programi iz časopisa ru javnom vlasništvu. Takođe, teoretski programi nastali, na primer, na PC računaru mogu se veoma malim izmenama raditi na PCW ili CP/M računarima. U praksi, jednom C programeru zasmetaće bi izvesna siromašnost u mogućnostima programiranja korisničkog interfejsa, a takođe nedostaje podrška raznim grafičkim adapterima. I kao da na PC računarima nema 640K centralne memorije, *Work-FORTH* radi samo u jednom segmentu od 64K, kao nekada *BASICA*. Takođe nema spomena ni o proširenjima memorije po LIM EMS, EEMS ili drugim standardima.

Dakle, ako već znate fort, ili ako želite da vidite šta su drugi našli u njemu tokom svih ovih godina — *Work-FORTH* može biti od koristi. Kao i svaki drugi fort, i *Work-FORTH* je dobar za pisanje drajvera za novi (na primer, eksperimentalni) hardver. Ali u konkurenciji drugih, bolje opremljenih i podržanih programskih jezika (razne varijante C-a, *JPI Modula-2*, *Turbo Pascal 4.0* itd.) — *Work FORTH* nema šta da traži kao sredstvo za razvoj komercijalnih ili aplikativnih programa.

*Duško Savić*

Komercijalni softver

Integrisani paketi / DIAMOND

# Dijamantski rez

**Većina programa piše se sa ciljem da pokrije jednu strogo definisanu vrstu potreba: računarsku grafiku, pisanje i obradu teksta, proračuna dinamičkih tabela — a namenjeni su placima i sekretaricama, inženjerima i crtačima, direktorima i knjigovođama. Specijalizovani programi su dugo dominirali ponudom softvara iz nekoliko jasnih razloga: svaka od pomenutih grupa predstavlja jasno izdvojano tržište, prvi popularni mikroračunari („epl II", "CP/M standard) bili su skromnih kapaciteta i nisu se mogli koristiti za zadovoljavanje raznorodnih zahteva u isto vreme, itd. Onda se pojavio *Lotus 1—2—3*...**

Za razliku od specijalizovanih programa *Lotus 1—2—3* je prvi u jednom programu ponudio proračune tabela (spreadsheet) grafiku i osnove rada sa bazom podataka. Softverske kuće su dobro shvatile da postoji tačno određena i platežno veoma sposobna ciljna grupa menadžera i direktora kojoj su potrebni i pisanje izveštaja i grafički prikaz rezultata poslovanja i proračun troškova pa čak i prikupljanje najnovijih podataka sa berze u realnom vremenu. Tako je rođena nova kategorija programa — integrisani paketi!

Samom svojom rasprostranjenošću *Lotus 1—2—3* je postavio standard za dinamičke tabele pa danas i nema programa koji ga ne bi imitirao u manjoj ili većoj meri. Zato je u srcu svakog integrisanog paketa dinamička tabela (spreadsheet) a procesor reči grafika, telekomunikacije ostaju u većini slučajeva na nivou pošlepalica. Od velikog broja integrisanih paketa svega nekoliko (*Symphony, Framework, Smart*) odstiglo je neku veću popularnost. U *Framework*-u su sastavni delovi (proračuni tekst-procesor poslovna grafika telekomunikacije, baza podataka *mailmerge* sakiciranje sadržaja dokumenta ugrađeni programski jezik makro naredbe) podjednako dobri kako jedan u odnosu na drugi tako i u odnosu na specijalizovane programe iste namene. Ostali integrisani paketi kao najjači deo imaju proračun dinamičkih tabela, zatim grafiku obrada teksta je po pravilu relativno slabe a telekomunikacije makro-naredbe i ostale pogodnosti rezervisane su samo za najveće proizvođače softvera. Program *Diamond* ističe se između dvadesetak integrisanih paketa. On nudi sve najvažnije osobine *Lotus*-a *1—2—3* procesor reči se skoro može meriti sa specijalizovanim programima iste namene a poslovna grafika je odlična. Slika i rezultati mogu se umetati u tekst — kao poručeno za poslovne ljude. Doda li se svemu tome i relativno niska cena jasno je zašto ovaj program postaje sve popularniji



## Ambicije u praksi

Osnovni cilj *Diamond*-a je da nastavi tamo gde je *Lotus 1—2—3* stao jer brza izračunavanja dinamičkih tabela nisu sama sebi cilj njihove rezultate treba obično ilustrovati i opisati nekim tekstom. Zato je *Diamond* pre svega *1—2—3* kompatibilan što znači da se direktno mogu učitavati datoteka programi iz *Lotus*-a *1—2—3*. *Diamond* zatim nudi kompletan procesor reči podržan odličnim *Miriam -Webster* datotekama za proveru spelovanja i pronalaženje sinonima (**tezaurus**) Po ugledu na uzor i *Diamond* može od delova tabela da pravi grafički prikaz, i — štaviše — može tako dobijenu sliku da uvaze u svoj procesor reči i odštampa je zajedno sa tekstom i rezultatima iz tabela. Poslovna grafika je sasvim kompletna na raspolaganju je čak 17 tipova dijagrama i 11 vrsta slova za obeležavanje crteže. Drugim rečima, *Diamond* je ambiciozan program i evo kako to izgleda u praksi

Dva su osnovna načina za komuniciranje sa ovim programom direktne naredbe (premeštanje kursora ili promena formule) i biranje iz menija. U svakom trenutku pritisak na kosu crtu / vodi u neki meni a **Esc** iz submenija prelazi u meni višeg nivoa. **Break** vodi u meni najvišeg nivoa za datu programsku celinu (tabelu procesor reči ili poslovnu grafiku). Taster **F1** u svakom trenutku prikazuje **Help** ekran, i to zavisno od situacije u kojoj je pomoć tražena *(context-sensitive help)*

Osim običnih naredbi korisnik može da pravi makro-naredbe i to identične onima u *Lotus*-u *1—2—3*. To je bitno, jer za *Lotus 1—2—3* postoje stotine već pripremljenih grupa makro-naredbi (što po knjigama što kao nezavisni softverski paketi) i mogu se lako nabaviti i koristiti u *Diamond*-u

Makro naredbe uglavnom imaju smisla samo u dinamičkim tabelama ali su tu i najpotrebnije. Korisnik može da pravi sopstvene menije, a pojedine makro-naredbe se mogu pozivati kao potprogrami. Postoji mogućnost da se izvrši makro-nareda odmah po učitavanju programa, što je veoma pogodno za rutinske kancelarijske poslove. Od velike je pomoći i mogućnost da se makro-naredbe izvršava korak po korak.

Glavni meni je jednostavan **Spreadsheet** (dinamička tabela) **Word** (prelazak u procesor reči) **Enter—DOS** (odlazak u DOS) **Hardware** (instaliranje periferijskih uređaja) **Colors** (podešavanje boja na ekranu) i **Quit** (napuštanje programa). Korisnik će najveći deo vremena provesti u opciji **Spreadsheet**, pa se grafikoni formiraju iz nje a ne iz glavnog menija.

## Dinamička tabela

*Diamond* ima dva režima rada **Menu** (biranje opcija iz menija) i **Ready** (unos podataka). Postoje tri vrste kursora osvetljeno polje u dinamičkoj tabeli, pulsirajući kursor koji se javlja u toku promene ili unosa podataka, i kursor u meniju koji osvetljava celo ime opcije. Kursori se kreću standardnim tasterima sa numeričkog dela tastature

Ekran je podeljen na dva dela. U gornjem su podaci (ukupno 21-na linija) a donje tri linije su rezervisane za status programa, prikazivanje submenija i konverzaciju sa korisnikom. Gornji deo je po svim osobinama i izgledu sličan *Lotus*-u *1—2—3* s tim što su dimenzije veće do 255 stubaca i 8192 reda.

Osnova za korišćene dinamička tabela je pojam oblasti *(range)*. Ona unosi dodatnu strukturu u inače amorfnu matricu dinamičke tabele. Oblast je uvek pravougaona tako da se zadaje gornjim levim i donjim desnim elementom. Zadavanje dimenzije oblasti izvodi se na dva načina unošenjem lokacija dvaju elemenata tabele, ili pokazivanjem na ta dve elemente pomoću specijalnog markera. Oblast se može imenovati što bitno utiče na ugodnost pri radu. Formule se zadaju kao i u svim ostalim programima ove vrste, tako da tu nema iznenađenja. S obzirom da radna oblast može biti veća od ekrana *Diamond*

### Tehnički detalji

*Diamond* je napisala i prodaje firma „Mosaic Marketing" (1972 Massachusetts Avenue, Cambridge MA 02140 USA). Košta 195 dolara što je u odnosu na konkurenciju sličnih mogućnosti — bagatelno. Sastoji se iz pet disketa od 5.25 inča (System Disk 1 System Disk 2 Dictionary, Thesaurus, Help/Library) ili tri diskete formata 3.5 inča. Priručnik je knjiga od 400 strana. Veoma je čitljiv i program se može savladati za svega nekoliko sati (pod opravdanom pretpostavkom da korisnik već ima iskustva sa dinamičkim tabelama i procesorima reči). Tekst je prilagođen i početnicima i veteranima ali se ipak svima preporučuje da rad sa programom započnu čitanjem glave 5 *(Tutorial)*. Na disketi Help/Library isporučeni su primeri koje treba prorađivati u toku čitanja *Tutorial*-a tako da je učenje u velikoj meri olakšano

*Diamond* se izvršava na računarima PC standarda sa dve disketne jedinice ili tvrdim diskom 320K centralne memorije ili više (preporučljivo je imati svih 640K) pod DOS-om 2.0 ili sa nekim drugim koji ima veći redni broj. Program se može koristiti i bez sledećih hardverskih dodataka monitor i grafička kartica u boji kartica sa RS—232 interfejs ploter (sa ne više od osam pera) štampač koji može da štampa grafiku, herkules grafička kartica sa monohromatskim monitorom, ali prisustvo svakog od njih mnogo povećava mogućnosti i komfor korisnika. *Diamond* ispituje da li je matematički koprocesor prisutan i koristi ga ako jeste

omogućava da se tabela na ekranu prikazuje u dva zasebna horizontalna ili vertikalna prozora. Izmena u jednom prozoru automatski menja sadržaj cele tabele, što se može a i ne mora odmah videti na ekranu.

Formule u *Diamond*-u su moćne. Adrese elemenata ili čitavih oblasti mogu biti zadate ili relativno (najčešći slučaj) ili apsolutno, a mešanjem apsolutnih i relativnih adresa moguće je, na primer, pomnožiti red sa redom. Uopštenje formula su funkcije, preciznije, funkcija je formula koja je sastavni deo dinamičke tabele. Funkcije uvek počinju znakom @, pa je tako @TAN(A6) tangens broja na adresi A6, a @SUM(B1..B25) je zbir svih brojeva u stupcu B, od prvog do 25-og elementa. Pomoću funkcija korisnik može da naznači koji će uslovi dovesti do prijavljivanja greške tokom proračuna, da ukaže da će vrednost elementa tabele biti izračunata u nekom kasnijem trenutku, da obavlja sva uobičajena aritmetičke operacije, statističke proračune nad nizovima brojeva, izračunava najčešće finansijske konstrukcije, pomoću vrednosti jednog elementa tabele bira neki drugi element, koristi logička operacija za poređenje i @IF naredbu, obrađuje obeležja stringova i slično. Funkcije se dele na specijalne, aritmetička, trigonometrijske, statističke, finansijske funkcije iz bora, logičke, vremenska, statistička u odnosu na neku bazu podataka definisanu u okviru tabela, kao i na funkcije za rad sa stringovima.

## Po prirodnom redosledu

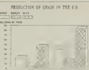
Glavni meni za dinamičku tabelu sadrži sledeće opcije: **Worksheet** (podešavanje opcija za celu tabelu ili za samo jedan deo), **File** (rad sa datotekama), **Range** (svi parametri oblasti), **Copy** (kopiranje oblasti), **Move** (premeštanje oblasti), **Graph** (poslov na grafika), **Data** (elementarne operacije nad bazom podataka u okviru dinamičke tabele), **System** (odlazak u DOS) i **Quit** (odlazak u glavni programski meni). Korišćenjem tih opcija i submenija koje one pobuđuju može se, na primer, odrediti da način predstavljanje brojeva bude u fiksnoj, naučnoj, poslovnoj ili nekoj drugoj notaciji i slično. Između mnogih ali ipak za ovu vrstu programa uobičajenih mogućnosti posebno se ističe opcija **Worksheet Global Recalc**. Naime, po definiciji, dinamička tabela se preračuna va posle unosa svakog novog podatka, ali postoji nekoliko načina da se to izvede.

U ovom programu proračun tabela se vrši u takozvanom prirodnom redosledu. To znači da — za razliku od većine drugih programa ovog tipa — *Diamond* razaznaje međuzavisnost formula, pa na osnovu toga prvo izračunava nezavisne formule, pa tek onda sve ostale. Time se izbegava proračun elemenata tabela čija vrednost nije inovirana. No korisnik ovde može da specifikuje i drugi način izračunavanja, na primer ručni. To znači da se tabela ne izračunava pri svakom unosu nego tek kada korisnik to eksplicitno naredi, što štedi vreme u velikim tabelama. Konačno, može se odrediti koliko puta treba tabele da bude iznova preračunata. To je neka vrsta petlje i veoma je zgodna ako formule opisuju neki iterativni proces. U tom postupku je moguće ući i u beskonačnu petlju, ali *Diamond* to može otkriti ako je odobren prirodni redosled izračunavanja.

Od drugih mogućnosti spomenimo da se redovi i kolone mogu brisati, premeštati, kopirati, da se širina kolona može menjati, da se naslovi nad kolonama mogu fiksirati kao i da se može u samu tabelu uneti znak za skok na novu stranu kod štampanja. Oblast može biti zaštićena od brisanja, sakrivena, a datoteka može biti stavljena pod lozinku. (Nije moguće prepoznati datoteku koja je na sličan način zaštićena u programu *Lotus 1-2-3*.)

Tabela na disku može se spojiti sa tabelom u memoriji, a deo tabele se može izdvojiti kao zasebna datoteka. *Diamond* može da čita datoteke u formatu **DIF** (praktički standard za razmenu podataka između *spreadsheet* programa) i naravno *Lotus 1-2-3 1A*, a u tim formatima može i da ih snima.

Štampanje je rešeno jednostavno. Može se (između ostalog) odštampati cela datoteka, cela tabela ili samo oblast, moguće je umetnuti zaglavlje i podnožje strenice, a pre početka štampanja poslati poseban niz upravljačkih znakova na štampač.




**Superbrzi histogrami: *Diamond* na delu**

## Procesor kao integrator

Rudimentarne operacije po ugledu na baze podataka moguće su i nad podacima u tabeli (opcija **Data**). Uopšteno govoreći, korisnik može da sortira podatke, izračunava distribuciju frekvencija u datoj oblasti, ispuni oblast nekom vrednošću, tabelira vrednosti promenljivog ulaznog podatka i bavi se sledećim operacijama: pretražuje tabelu kao bazu podataka, briše podatke, filtrira oblast po nekom kriterijumu, te jedinstvene fiksne primerke prebaci u neki drugi deo tabele. Ovde je najinteresantnija opcija za tabeliranje. Pretpostavimo da formula glasi @FV(1000,A2,4). Element **A2** je varijabla za koju treba zadati niz vrednosti, a *Diamond* će ih sačuvati u nekoj oblasti kao niz rešenja.

Postoje i naredbe za rad sa matricama (u matematičkom smislu): množenje i invertovanje matrica. Stoga je *Diamond* pogodan i za razne matematička i statistička obrada podataka.

Sve u svemu, dinamičke tabele u ovom programu su kompletno obrađene i tu *Diamond* ne ustupa mesto ni *Lotus-u* ni ostalim sličnim programima. Ali, za razliku od uzora, *Diamond* ima ugrađen i procesor reči.

U deo za obradu teksta ulazi se biranjem opcija **Word** iz glavnog menija. Pojavljuje se ekran sa sličnim rasporedom kao i u dinamičkoj tabeli: gornji deo je za tekst, donje tri linije sadrže meni. Tasterom Esc se dolazi do tog menija, a **Alt** u kombinaciji sa drugim tasterima stoji na raspolaganju kao ubrzani način biranja opcija iz menija.

### Instaliranje programa

Postupak je jednostavan. Korisnik najpre formira novi imenik na tvrdom disku, a zatim za svaku disketu ponavlja DOS naredbu COPY A:*.*. Program se startuje sa DD, a ako treba koristiti herkules grafiku, onda pre učitavanja programa treba aktivirati programe **SETFULL** (koji postavlja karticu u punu konfiguraciju) i HARDCOPY (da bi se pritiskom na PrtSc moglo na štampaču dobiti kopija ekrana). Po učitavanju *Diamond*-a pojavljuje se glavni programski meni. U njemu treba odabrati opciju **Hardware**, i zatim popuniti podatke o priključenim periferijskim uređajima. Definišu se vrsta ekrana i boja, štampač, ploter, brzina prenosa preko serijske kartice i parametri strane na štampaču (margine, dužine lista, itd). Podržani su sledeći štampači: Epson MX/FX/RX, IBM (Graphics B&W, Color, ProPrinter), Okidata 92, Dale Products 8050/8070 (IDS Prism), HP LaserJet, a moguće je navesti i da nema štampača. Podržani ploteri su HP7470 sa dva pera, ColorPro/HP (...)& svi ploteri koji se pridržavaju standarda HPGL, Strobe 260, EMUPLOT štampač, HI DMP-29/DMPL, a takođe se može navesti da nema plotera. Zastupljeno je osam standardnih brzina prenosa serijskom karticom, od 110 od 9600 boda.

Konfigurisanje periferija u samom programu ima tu prednost da se izmene mogu sprovesti na izlazeći iz programa — a to je u praksi najčešće i potrebno.

Sam način rada je prilično standardan za sve procesore reči: kursorski tasteri premeštaju kursor po ekranu, tekstovi mogu biti proizvoljne dužine (to jest, dužina teksta ograničena je samo prostorom na disku), a tu su i označavanje blokova, premeštanje, kopiranje, brisanje i tako dalje. Važne opcije i jake strana ovog programa — provera spelovanja i rečnik sinonima — primenljive su, naravno, samo na engleski jezik. Opcije za rad sa datotekama i štampačem pomalo su ograničene (na primer, proted može biti samo od jednog od tri reda), ali ipak sasvim standardna i upotrebljiva.

Drugim rečima, *Diamond* se može lepo koristiti za pisanje, a specijalizovani programi (*Word*, *WordPerfect* i ostali) ga dakako daleko nadmašuju po dodatnim mogućnostima. Iako je sasvim moguće, teško bi bilo poverovati da će neki profesionalni pisac koristiti *Diamond* za stvaranje tekstova, ali to nije svrha integrisanog paketa. Maksimum što se traži od programa ove vrste je pisanje poslovne ponuda, ugovora, odpisa, lak prenos brojeva iz neke tabele i lepa ilustracija poslovnih rezultata. *Diamond* to u potpunosti omogućava, pri čemu procesor reči služi kao integrator reči, brojeva i grafike.

## Nije za svakoga

Naredbom **Esc Transfer** se oblast prenosi iz tabela u tekst, a moguće je i obratno. Na primer, korisnik može procesorom reči da učita ASCII datoteku podataka iz programa *dBASE III*, da je premesti u dinamičku tabelu i tamo od nje napravi bazu podataka u formatu *Diamond*-a. I to nije sve. Opcijom Graph se u dinamičkoj tabeli može grafički predstaviti niz brojeva iz neke oblasti, i ta slika se može preneti u procesor reči. Slika se neće videti u okviru tekst procesora, ali će biti odštampana tačno tamo gde korisnik zaželi. Postoje svega tri veličine slike, tako da *Diamond* ni u kom slučaju nije program za stono izdavaštvo.

Program za poslovnu grafiku transformiše skup brojeva u sliku: najčešće grafikon „torte" histogram ili neki kombinovani oblik. Svoju ogromnu popularnost *Lotus 1-2-3* duguje pre svega brzini proračuna, ali i mogućnostima grafičke prezentacije poslovnih rezultata. Slobodno se može reći da je poslovna grafika u *Diamond*-u bolja u odnosu na druge slične programe. Korisnik može da pregleda rezultate, pa da se tek onda odluči za najbolji oblik predstavljanja. Osim toga, rezultati se mogu izdavati i na slajdovima, a to je ipak najbolji način prezentacije slika grupi ljudi, na primer potencijalnim poslovnim prijateljima.

Na raspolaganju je čak sedamnaest vrsta dijagrama: **X/Y osa** (jedna osa, dve ose, linearno-linearno, logaritamske, logaritamsko-linearne, logaritamsko logaritamske ose i slobodno raspoređivanje), linije (jedna i dve osa), **horizontalni histogrami** (pozitivni, pozitivni i negativni histogrami), **trodimenzioni histogrami, „torte", trodimenzione „torte", „torte" i histogram, i tekst.** Tekst ima 11 oblika slova (fontova), osam veličina i popriličan broj opcija za umetanje poruka u ilustracije.

Prvi korak u stvaranju slike je unošenje podataka u dinamičku tabelu. Zatim treba odabrati tip slike, definisati raspone podataka, upisati oznake stubića histograma, precizirati opcije (naslov, podnožje crteža, oznake osa, razmeru, boju, vrstu i veličinu slova, tip šrafure i legendu). Onda se crtež može prikazati na ekranu ili odštampati. No sve to ne znači da se mora svaki parametar potanko definisati unapred. Velika prednost *Diamond*-ove grafike je što se početka slika može dobiti i bez ulaženja u detalje, a zatim se postepeno podešavati prema ciljnoj grupi gledalaca. Proces kreiranja slike je u stvarnosti veoma jednostavan: označi se oblast u dinamičkoj tabeli i izvrši naredba **Graph**. Slika se čak i na XT u sa 4.77 MHz (ali uz koprocesor 8087) dobija skoro trenutno.

*Diamond* je bez sumnje odličan program. Ostale lekmace tuče cenom, brzinom, izvesnom grafikom, proverom spelovanja i tezaurusom, kao i potpunom usaglašenošću sa *Lotus*-om *1-2-3*. No daleko od toga da je ovaj program za svakoga. Namenjen je ekonomistima, knjigovođama, direktorima, savetnicima — svima koji žele da do prikaza svojih poslovnih rezultata i predloga dođu na relativno lak, brz i jednostavan način.

***Duško Savić***

Komercijalni softver
Asembleri/OPTASM

# U dosluhu s đavolom

**Mašinsko programiranje PC računara ulazi u novu eru, i to jednim krupnim, nezadrživim korakom. Revoluciju je izazvala kompanija „SLR Systems", svojim asemblerom za mikropocesore iz familije 86/286. Program nosi naziv *OPTASM* i potpuno je kompatibilan sa Microsoft-ovim makroasemblerom *(MASM)*, koji je do sada, suvereno vladao tržištem i radnim stolovima programera. Sudeći po spektakularnoj brzini, starom kralju su dani odbrojani. Tu, međutim, prednosti programa *OPTASM* tek počinju . . .**

Običaj je u zapadnom svetu da se svaki novi softverski paket prvo pošalje na takozvani „bete test" odabranim programerima i korisnicima sa ciljem da se utvrde svi eventualni bagovi ili da se isprave propusti koji možda u fazi izrade programa nisu uzeti u obzir. Tako je i *OPTASM* proveo par meseci u rukama vrhunskih poznavalaca asemblera, gde se kalio.

Reakcije su više nego burne. Pohvale stižu sa svih strana, a rezultati testova govore sami za sebe. Na AT-u pri kloku od 6MHz, OPTASM je 4 (i slovima četiri) puta brži od MASM-a verzije 5.0. Brzina asembliranja se procenjuju na 75000 linija u minuti što je podatak od koga zastaje dah.

Jedan od onih koji su program testirali, predsednik kompanije „Penscope" Biel Seller, uz sve komplimente programu izražava samo željenje što više ne može da skokne na kafu dok traje asembliranje, dok jedan drugi ljubitelj asemblera ne može da poveruje da je živ čovek napisao takav asembler i izražava sumnju da je programer *OPTASM*-a sklopio pakt sa — samim đavolom.

## Viteški turnir

Sve ove lepe priče stigle su u našu redakciju zajedno sa programom. Prihvatili smo ih sa interesovanjem ali i sa izvesnom dozom skepse, sve dok program nismo startovali.

Brzina je zaista fascinantna. Tu brojevi ne mogu ništa da dočaraju. Subjektivan doživljaj je daleko drugačiji. Treba jednostavno propustiti neki izvorni tekst kroz *MASM* pa kroz *OPTASM* i sve će biti jasno. To se ne može ni porediti.

Odnos brzina 4 je uglavnom u redu, ali naše je mišljenje da može i bolje. Pre svega, ako se radi sa programskim modulima, onda primena „MAKE" datoteke može dodatno ubrzati stvar (o „MAKE" datotekama reći ćemo nešto više kasnije). I nesevno svuda je tu uračunato i značajno vreme potrošeno na komunikaciju sa diskom. Mi smo odlučili da testiramo iz RAM diska tek da bi neke stvari postale jasnije. Korišćen je XT kompatibilac pri kloku od 10MHz. Probni izvorni tekst imao je oko 60K i bio je „obogaćen" raznoraznim naredbama za skok, specijalno za tu priliku.

*MASM 4.0* obavio je posao za 53 sekunde i proizveo objektni kod dužine 13K. Nešto bolje je prošao *MASM 5.0*: 42 sekunde, uz istu dužinu objektnog koda. Konačno, *OPTASM* je preveo program za neverovatnih sedam sekundi, a objektni kod je zauzeo tek 11K.

Komentar je svakako suvišan.

## Optimizacija koda

Skraćenica „*OPTASM*" potiče od „Optimizing Assembler" čime se naglašava osobina asemblera da proizvodi optimiziran, odnosno najkraći mogući objektni kod. Na prvi pogled može se tu postaviti pitanje „Kakva sad optimizacija, kada je mašinski jezik sam po sebi optimalan — svaka mnemonička oznaka odgovara jednoj mašinskoj naredbi?" Ali, verovali ili ne, optimizacija je moguća. Naravno, nije to ništa ni približno tako zamašno kao što je



prevođenje fortrana ili paskala, ali posla ima.

Uzmimo, na primer, naredbu za bezuslovni skok tipa **JMP neka_adresa**. Objektni kod za takvu naredbu može da ima dva ili tri bajta, zavisno od toga koliko daleko se nalazi **neka_adresa**. Ako je blizu, **JMP** se prevodi u dva bajta kao relativan skok u opsegu od 128 bajtova napred ili nazad. Za duže skokove **JMP** se prevodi u tri bajta.

Izgleda prosto, ali asembleru baš i nije tako lako da proceni da li je skok kratak ili dug. Pokazaćemo to na primeru opštepoznatog i opšteprihvaćenog **Microsoft-ovog** makroasemblera, pod nazivom MASM.

## Dva prolaza . . .

Klasični asembleri, u koje spada i MASM, prevode program u dva prolaza.

U toku prvog prolaza MASM čita izvorni tekst liniju po liniju i svakoj naredbi rezerviše odgovarajući broj bajtova za njen objektni kod. Za neke naredbe, kao što je recimo **XOR AX,AX** može se odmah generisati i objektni kod ali za neke druge, ko što je **JMP neka_adresa**, to ne mora da bude slučaj jer pitanje je da li asembler zna gde je ta **neka_adresa**. Sasvim je moguće (i vrlo često se i događa) da se neki simbol, na koji se tekuća naredba poziva, nalazi negde kasnije u listingu i do njega će asembler tek doći. Zato *MASM* u prvom prolazu i ne generiše objektni kod već, kao što smo to rekli, samo broji bajtove i rezerviše prostor. Naredbi tipa **JMP neka_adresa** dodeliće tri bajta, za svaki slučaj, izuzev ako se pseudonaredbom SHORT ne naglasi da je skok kratak.

Osim toga, važan posao u toku prvog prolaza je i prikupljanje svih simbola, i njihovo smeštanje u posebnu tabelu. Pored svake simboličke adrese upisuje se odmah njena stvarna vrednost, što će asembleru biti potrebno kasnije.

U drugom prolazu, asembler generiše objektni kod. Kada bude naišao na naredbu JMP neka_adresa, *MASM* će pročešljati onu tabelu simbola formiranu u toku prvog prolaza, i uzeti odatle adresu za skok. Ako je skok duži od 128 bajtova, sve tri polja rezervisana za objektni kod biće iskorišćena. Međutim, šta ako je skok kraći? Za objektni kod su dovoljna dva bajta. Ali nije baš jednostavno izbaciti suvišan bajt, jer bi to poremetilo sve kasnije adrese. Zato *MASM* bez mnogo razmišljanja, zadržava i taj treći bajt a u njega upisuje vrednost 90 heksadecimalno, što je objektni kod mašinske operacije NOP. Problem je tako rešen. Ne baš naročito elegantno, ali ipak će raditi. Uostalom, taj NOP se nikada neće ni izvršiti, pošto stoji neposredno iza naredbe za bezuslovni skok.

Naravno, nije to samo stvar elegancije. Ako je program veliki sam po sebi, onda će i broj ovakvih veštačkih NOP-ova biti veliki, što će predstavljati najobičniji memorijski teret. A memorije obično nikad dosta.

## Ili n prolaza.

S druge strane imamo *OPTASM* koji prevodi program u „n" prolaza, gde to „n" može biti proizvoljno veliko, sve dok se ne dobije optimalan kod. Najinteresantnije je to što uvećan broj prolaza uopšte ne uvećava vreme prevođenja, već ga, naprotiv, umanjuje! Treba videti da bi se verovalo.

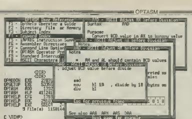
Optimizacija se vrši i u nekim drugim naredbama, a program se čak po potrebi i modifikuje da bi se željeni rezultat postigao. Najbolji primer za to su uslovni skokovi, koji inače nikako ne mogu biti duži od 128 bajtova (svako ko se bavi mašinskim programiranjem mikroprocesora 8088 ovu tužnu činjenicu i te kako dobro zna). Pokušaj skoka preko ove granice MASM neće prihvatiti, već ćemo morati da izmenimo našu neaačnu naredbu, koja je, recimo, glasila **JNZ EXIT**, i staviti umesto nje nešto poput:

JZ CONTINUE
JMP EXIT
CONTINUE: nastavak programa

*OPTASM* ovakve stvari radi sam i mi mu tu nismo potrebni. Jedino ukoliko baš izričito ne želimo da nam asembler vršlja po programu, možemo to naglasiti — i program će nam ostati neizmenjen. A pokušaj dugih uslovnih skokova završiće se prijavom greške, baš kao da je u pitanju stari dobri *MASM*.

## Fazne greške

Ako radimo sa makroasemblerom (a radimo sigurno jer za druge asemblere nismo ni čuli), onda zasigurno znamo šta je to *Phase error*, odnosno fazna greška. Viđamo je prilično često na ekranu i uvek se iznenađeno pitamo „Pa zašto sad?" E pa zato što *MASM* nije savršen asembler i kao takav jako teško izlazi na kraj sa takozvanim *forward* referencama, odnosno pozivanjima na simbole koji će biti definisani tek kasnije u listingu. U istu grupu problema spada i onaj NOP u okviru naredbi za bezuslovni skok, ali ima situacija kad ni NOP ne pomaže




**U stilu *Norton Guida* izvanredni rezidentni HELP pritiskom na nekoliko tastera daje sve podatke o korišćenju asemblera i instrukcijama procesora 8086/286, uključujući i primere upotrebe**

i asembler tada prosto diže ruke od daljeg prevođenja programa

Uzećemo jedan jednostavan primer

```
        CALL    TABLE
        MOV     AH 4CH
        INT     21H
SUBROUT XOR     AX AX
        MOV     ES.[BX],AX
        RET
TABLE   DW      SUBROUT
```

Ovaj, na izgled korektan niz instrukcija, *MASM* neće umeti da prevede. Prijaviće nam grešku „Phase error between passes" Međutim ako **CALL TABLE** premestimo na kraj niza (iza definicije simbola TABLE) problema neće biti

*OPTASM* ne pravi probleme ni u jednom ni u drugom slučaju. Fazne greške su stvar prošlosti

## Instalacija i korišćenje

Program OPTASM (verzija 1 0) dobija se na samo jednoj disketi što možemo okarakterisati kao prijatno iznenađenje imajući u vidu danas sve prisutnije (megalomansku?) tendenciju širovanja softvera na po desetak i više disketa. Obimno uputstvo koje se uz program dobija, veoma je kvalitetno i rekli bismo kompletno. Na oko tri stotine strana dato je sve što treba jednom korisniku asemblera. Pažnja je posvećena i onim malo iskusnijim programerima, koji su već dobar deo svog vremena, iskustva (i nerava) uložili u stari *MASM*. Za njih se na posebno uočljiv način tokom čitavog teksta izdvaja samo ono što je karakteristično za *OPTASM*, a što *MASM* nije imao. Lepo pregledno i korisno

Osnovna datoteka na disketi je jasno OPTASM EXE. Tu je i par pomoćnih programa za instalaciju, odnosno konfigurisanje asemblera. Po volji se mogu menjati svi parametri značajni za rad programa, i kada se to jednom uredi, kasnija upotreba asemblera je veoma jednostavna

Komandna linija može imati kratak oblik, kao recimo

OPTASM probe,

Po potrebi, mogu se u komandnu liniju uneti i razne opcije, na uobičajen način (korišćenjem kose crte). Ove opcije neposredno utiču na rad asemblera u istoj meri koliko to rade i pseudoinstrukcije u okviru izvornog teksta

Višestruko je ubrzan rad sa programskim modulima. Ako recimo pišemo neki program koji se zove *ACCOUNT*, a sastoji se iz tri zasebna modula sa imenima *MAIN CALC* i *REPORT*, onda će, verovatno, svaka izmena u jednom od tih modula zahtevati i izmene u ostalima. I svaki put ćemo morati da izvršimo niz od tri poziva asemblera

OPTASM MAIN
OPTASM CALC
OPTASM REPORT,

Ne samo što je zamorno za kucanje već i izvršenje dugo traje. (Što se tiče kucanja tu bi pomogla jedna BAT datoteka, ali na brzini se ništa ne bi dobilo)

Umesto toga, može se formirati jedna ASCII datoteka, sa sledećim sadržajem

MAIN,
CALC,
REPORT,

To će biti takozvana „MAKE" datoteka a nazvaćemo je recimo ACCOUNT MAK i nadalje ćemo naše tri modula odjednom asemblirati naredbom

OPTASM @ACCOUNT MAK

Prednost ovakvog postupka nad primenom BAT datoteke je u tome što se sada asembler neće tri puta učitavati sa diska. Učitaće se samo jednom a asembliraće sa redom moduli iz datoteke ACCOUNT.MAK. Znak „@" stoji ispred imena samo da bi naznačio asembleru da ACCOUNT MAK nije ime izvornog teksta nekog programa, već ime „MAKE" datoteke

## Deklaracije segmenata

Asembler *OPTASM* ne zahteva da izvorni tekst programa započne ceremonijalnim deklaracijama segmenata, mada su na ovu torturu korisnici makroasemblera već odavno navikli, pa su čak našli da ona doprinosi preglednosti programa. Naravno, ništa neće smetati ako deklaracije u listingu i ostanu, kao na primer

```
TITLE   Pozdravna   poruka
STACK   SEGMENT  PARA STACK
                 'STACK
        DB       256 DUP(0)
STACK   ENDS
DSEG    SEGMENT  PARA PUBLIC DATA'
MESSAGE DB       Pozdrav svima!$
DSEG    ENDS
CSEG    SEGMENT  PARA PUBLIC CODE
        ASSUME   CS CSEG DS
                 DSEG SS STACK
POZORAV PROC     FAR
        MOV      AX DSEG
        MOV      DS AX
        LEA      DX MESSAGE
        MOV      AH,09H
        INT      21H
        MOV      AH 4CH
        INT      21H
POZORAV ENDP
CSEG    ENDS
        END      POZORAV
```

Ali, sasvim lako će proći i našto popul

```
TITLE   Pozdravna poruka (uprošćeno)
        DOSSEG
         MODEL    SMALL
         STACK    256
         DATA
MESSAGE  DB       'Pozdrav svima!$
         CODE
POZORAV  MOV      AX,DGROUP
         MOV      DS AX
         LEA      DX MESSAGE
         MOV      AH 09H
         INT      21H
         MOV      AH 4CH
         INT      21H
         END      POZDRAV
```

Radi se o istom programu ali u obliku koji koristi uprošćenu deklaraciju segmenata, pomoću posebnih pseudoinstrukcija **STACK, CODE** i **DATA**. S druge strane određivanje tipa i rasporeda segmenata vrši se komandama **DOSSEG** i **MODEL**. Ideja je, zapravo preuzeta od kompajlera za više programska jezika gde korisnik nikada i nije neposredno birao i deklarisao segmente već je samo zadavao takozvani *memorijski model*, odnosno veličinu programa i podataka. Upravo u tu svrhu OPTASM uvodi instrukciju **MODEL**

Mogući tipovi memorijskih modela su sledeći

*TINY* — program i podaci zajedno staju u 64K,

*SMALL* — program i podaci ponaosob staju u po 64K

*MEDIUM* — podaci staju u 64K, ali program može biti i duži,

*COMPACT* — program staje u 64K, ali podaci mogu biti i većeg obima i

*LARGE* — ni program ni podaci ne staju u 64K

U zavisnosti od izabranog memorijskog modela, asembler će interno generisati nedostajuće SEGMENTE i ASSUME komande bez potrebe da one eksplicitno figurišu u listingu

## Prenošenje simbola

Kada pišemo veći program obično ćemo celinu razbiti na module koje razvijamo i testiramo zasebno a povezujemo tek u poslednjoj fazi linkovanja. Asembler će bez problema obrađivati svaki modul posebno onako kako ih mi budemo razvijali. Poteškoće nastaju samo onda kada u nekom od modula treba koristiti simbol definisan u drugom modulu

*OPTASM* tu ne izmišlja ništa novo. Problem se rešava deklaracijama **PUBLIC** i **EXTRN**, na uobičajen način. To znači da svi simboli u okviru jednog modula koji će biti korišćeni od strane drugih modula moraju biti navedeni u listi naredbe **PUBLIC**. Isto tako svi simboli koji se pozivaju iz drugih spoljašnjih modula moraju se navesti u listi naredbe **EXTRN**.

Međutim *OPTASM* pruža mogućnost jednostavnog deklarisanja svih simbola kao **PUBLIC**, ukoliko se samo pri asembliranju naznači opcija /G. Pri tome sama deklaracija **PUBLIC** u izvornom tekstu nije potrebna

## Lokalni simboli

Sve će programere sigurno obradovati mogućnost da definišu lokalne simbole koji će važiti samo u okvirima jedne procedure. Ako ni zbog čega drugog, onda zbog toga što nije ni malo prijatno izmišljati nova imena simbola i paziti da pri tome neko ne bude dupliran. A lako bi lepo bilo nazvati svaku programsku petlju jednostavno LOOP

Problem je rešen na sledeći način

Svaka labela koja počinje znakom # smatra se lokalnom, i njeno važenje prestaje kad se napusti procedura u kojoj je labela definisana

Svaka labela koja počinje cifrom a završava se sa $ takođe se smatra lokalnom. Na primer, lokalna su labele

#LOOP
1LOOP$

Pozivanje na ove simbole iz okvira drugih procedura nije moguće i ništa neće smetati ako se i u tim drugim procedurama nalazi lokalna labela sa istim imenom #LOOP

***

Teško je u jednom tekstu navesti sve osobine jednog softverskog paketa. Ali to nije ni potrebno da bi se o programu stekla prava slika. Cilj nam je bio da pre svega upoznamo naše čitaoce sa jednim proizvodom koji će izmeniti budućnost sveta PC računara. A vreme će kao i uvek učiniti svoje

*Jovan Skuljan*

## OPTASM

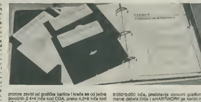
Programski paket *OPTASM* verzija 1 5 objavljena u avgustu koji obuhvata dve diskete i detaljan priručnik povezan spiralnom žicom može se poručiti od izdavača po ceni od 125 dolara + 10 dolara za poštanske troškove na adresu *SLR Systems, 1622 N Main Street Butler PA 16001, USA*

Komercijalni softver
CAD/smARTWORK

# Elektronski crtač

**Projektovanje elektronskih uređaja nije samo unosno zanimanje talentovanih inženjera nego i izuzetno rašireni hobi među mladima širom sveta. Najteži i najmukotrpniji deo u ovom poslu predstavlja, bez sumnje, projektovanje i crtanje štampanih kola. Računari su, na sreću, i ovde mnogo toga promenili — skupe i nedostupne crvena i plava termo stabilna „bišopove" trake danas predstavljaju samo ružan san, a dani i nedelje naizvesnog i napetog crtanja svedeni su na nekoliko časova prijatnog rada sa izvesnim ishodom. Na profesionalnom tržištu softvera postoji desetak programa za crtanje štampanih kola, ali je samo jedan među njima uspeo da osvoji podjednake simpatije i među amaterima i među crtačima od zanata. *SmARTWORK* nije savršen program. Njegove vrline su, međutim, toliko jake da mu korisnici rado opraštaju sve nevažna mane.**

Program *smARTWORK* (izuzetno uspela kovanica reči *smart* pametan i *artwork*, crtež) namenjen je za projektovanje jednostranih i dvostranih štampanih kola veličine do 10•16 inča (254•406 mm). Vodovi se mogu povlačiti ručno ali se posao može prepustiti i programu — *smARTWORK* će ih u saradnji sa crtačem crtati ili liniju po liniju ili će ih, u automatskom režimu, nacrtati sve odjednom. Ugrađeni autoruter program za automatsko izvlačenje veza, u drugom slučaju određuje najbolju strategiju crtanja. Po završenom crtanju, *smARTWORK* omogućuje štampanje kontrolnih crteža i suptilno doterivanje finalnog crteža na matričnom štampaču ili ploteru u svim varijantama za automatsku izradu štampanog kola. Izvan osnovnog paketa mogu se naručiti i drajveri za optički ploter i automatsku bušilicu za ruše. Projektovanje štampanog kola nije nimalo trivijalan zadatak — ono zahteva od programa priličnu dozu pameti. Stoga je *smARTWORK*, kao i većina savremenih paketa, pre program za automatsko crtanje nego za automatsko projektovanje. U program je, međutim, ugrađeno dovoljno inteligencije da se postigne visoka automatizacija mehaničkih poslova u crtanju i — da paket opravda ime koje nosi.



## U radnom prostoru

Rezultat rada *smARTWORK*-a je krajnje profesionalan i ne čudi što je program izuzetno prohtevan i osetljiv na hardversko okruženje. U najskromnijim varijantama *smARTWORK* može da radi na standardnom XT-u sa CGA grafičkom karticom, kolor monitorom i jeftinim „epson" kompatibilnim štampačem, pa čak i sa softverskim simulatorom CGA kartice na „herkulesu" ali je tada utisak o programu, mada je rezultat sasvim upotrebljiv u amaterskoj praksi, prilično bled. Na AT-u sa EGA karticom program postaje nešto sasvim drugo, jako se povećana rezolucija ne koristi za kvalitetniji prikaz na ekranu već za veće zahvatanje u radni prostor.

Za dvostrano štampano kolo *smARTWORK* upotrebljava dve radne površine — crteže spojeva i veza na gornjoj i donjoj strani ploče i crtež montažne šeme za takozvano „belo sito" — i prikazuje ih istovremeno na ekranu. Programi za projektovanje štampanih kola nisu po pravilu predviđeni za rad sa crno-belim grafičkim karticama jer istovremeni prikaz tri crteža ima smisla jedino ako su veze označene različitim bojama. *SmARTWORK* se u izboru grafičkih kartica dosledno pridržava ovih pravila ali uz jedan izuzetno praktičan izuzetak — crtač u toku rada može slobodno da bira između crno-belog prikaza samo jednog crteža (aktivne radne površine) u visokoj rezoluciji i prikaza u boji kompletnog štampanog kola. Sa CGA karticom mogu se birati dva osnovna tipa boje — crvena, zelena i žuta ili magenta, cijan i bela — dok EGA i VGA dozvoljavaju slobodno kombinovanje bilo koje tri boje iz palete od 16 osnovnih boja.

Svaka radna površina je presvučena vidljivom ili nevidljivom mrežom gustine 50 stotih delova inča (1,27 mm) na koju se nanose osnovni grafički elementi u oblikovanju štampanog kola. Ekran monitora predstavlja prozor u radnu površinu. Veličina ovog prozora zavisi od grafičke kartice i kreće se od jedva dovoljnih 2,4•4 inča kod CGA, preko 4,2•6 inča kod EGA do 6•6 inča kod VGA. Prozor se po radnoj površini pomera kursorskim tasterima kad se kursor dovede na neku od njegovih ivica. Program se uvek budi sa kursorom u donjem levom uglu.

U oblikovanju štampanog kola *smARTWORK* koristi četiri osnovna grafička elementa — tanku liniju, debelu liniju, stopicu i popunjeni kvadratić. Signalne veze se izvlače tankim linijama debljine 12, 16 i 20 stotih delova inča. Kombinovanje različitih debljine u tankim linijama na jednom na štampanom kolu nije dozvoljeno. Vodovi za napajanje se izvlače debelim linijama. One su široke 50 stotih delova inča i debljina im se može proizvoljno uvećavati umnoškom od 0,050 inča. Tanke i debele linije se mogu vući uspravno, vodoravno i koso pod uglom od 45 stepeni, a njihovim kombinovanjem grade se složene linije. Spojna mesta se crtaju okruglim ili četvrtastim stopicama prečnika 52 ili 75 stotih delova inča. Stopice su jedini trodimenzionalni grafički element u ovom paketu — imaju oblik kalema i njima se veza prenosi sa jedne na drugu stranu štampanog kola. Stopica se, postavljanjem na jednu, automatski postavlja na obe strane. Popunjeni kvadratić dimenzija 0,050•0,050 inča, predstavlja osnovni gradivni element debela linija i *smARTWORK* ga koristi na nekoliko načina. Između ostalog i za crtanje spojnih mesta za elemente sa površinskom montažom. Dimenzije tankih linija i stopica nemaju odraza na njihov prikaz na ekranu — one dolaze do izražaja tek prilikom crtanja.

### Nove verzije

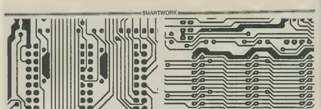
[illegible] *smARTWORK* je jedan od najpoznatijih [illegible] — proizvođač izbacuje [illegible] osam meseci. Sve izmene [illegible] u „Tehničkom biltenu" [illegible] nekoliko puta. Izmene [illegible] bile beznačajne, a sonde [illegible]. Poslednja značajna izmena [illegible], najnovija verzija je [illegible] VGA, a u pripremi je rutina [illegible] postavljanje i razmeštanje kompo[illegible] crtanja. *smARTWORK* [illegible] sve tužine moći samostalno i [illegible] kola. Bez obzira na sve [illegible] je očigledno još uvek neop[illegible] programa.

## Upravljanje programom

Program *smARTWORK* je podeljen na nekoliko funkcionalnih celina kojima se upravlja iz DOS-a. Pored programa za crtanje *EDIT* i programa za štampanje *DOT* i *PLOT* u paketu se nalazi i izvestan broj pomoćnih programa kojima se poveravaju specijalni zadaci. Ovakva koncepcija rasterećuje osnovni program specijalnih opcija, pojednostavljuje i ubrzava upravljanje i olakšava naknadnu obradu datoteka. U *EDIT* je ugrađen poseban komandni procesor, a pomoćnim programima se upravlja navođenjem odgovarajućih parametara uz ime programa prilikom njegovog poziva. *SmARTWORK* je međutim jedan od retkih programa koji ne poštuje uobičajena pravila za ovakvu vrstu upravljanja — umesto standardne kose crte (/) parametri se označavaju crticom (—), datoteka u koju se smeštaju izlazni rezultati i slovom o (—o), a izlazna i odredišna datoteka se navode u obrnutom redosledu.

U *smARTWORK* je ugrađen klasičan komandni interpreter i rukovanje programom je trivijalno jednostavno — jedno čitanje priručnika i tri lakonski sroče na HELP ekrana sasvim su dovoljna za lako crtanje. Uz grafičke naredbe i naredbe za manipulisanje štampanim kolom program poznaje i elementarne naredbe za rad sa datotekama — promenu i prikaz radnog kataloga, uzimanje i odlaganje crteža i liste veza sa diska i na njega i napuštanje programa. Skroman izbor prati na žalost i izuzetno kruta postavka — naredbe se koriste samo u svom elementarnom obliku i ne mogu se automatski ponavljati.

Projektovanje i crtanje štampanog kola je mukotrpan posao čak i kada se to radi *smARTWORK*-om i stoga je razumno crtež često odlagati na disk. Autori programa, međutim, ne pokazuju dovoljno razu-



Princeza od žabe Svoj pravi oblik štampano kolo dobija tek u procesu štampanja ili crtanja — podsecanje stopica, obrada spojnih mesta i popunjavanje vodove obavljaju se potpuno automatski, levo je detalj crteža u razmeri 2:1 na matričnom štampaču, a desno na ploteru

## Praktična iskustva

SmARTWORK se može koristiti za crtanje izuzetno složenih štampanih kola. U Institutu „Mihajlo Pupin" korišćenjem ranijih verzija ovog programa nacrtano je štampano kolo za „Tim 011", bez obzira što program ne podržava raster na nožicama procesora HD 64180 u DIL pakovanju. Za potrebe ovog testa u *smARTWORK*-u je projektovano i nacrtano štampano kolo punog formata upravljača diska sa baterijskim napajanjem za PC XT/AT sa 36 integrisanih kola. U ocenjivanju programa imali smo u vidu iskustva u radu sa programima *OrCAD* (crtanje shema), P-CAD (integrisani paket za projektovanje štampanih kola) *Tango* (crtanje štampanih kola) i *HiWire* (integrisani paket za projektovanje štampanih kola).

mevanja za ovu elementarnu potrebu — crtač svaki put mora da prelazi u komandni režim, kuca puni oblik naredbe i puni naziv crteža, grickajući nokte dok program mulje po disku. Crteži se čuvaju u izuzetno kondenzovanom formatu (po kartica punog formata uzima napunih 36 K!) i zato je njihovo odlaganje i uzimanje sa diska izuzetno sporo. Čim se crtač udalji od stopica, podnožja i vodova, čak i kada treba nacrtati tako trivijalnu stvar kao što je okvir štampanog kola ili komponente na crtežu za belo sito, *smARTWORK* postaje krut i neprijatan. Program je, uz to, potpuno zatvoren — u njega se ne može ući ni na koji način da bi se prilagodio stilu rada korisnika, i dobro zaštićen — može se preneti na tvrdi disk ali odbija da radi bez originalne diskete u drajvu A. *SmARTWORK* je pisan za hardveraše, tvrde i odvažne momke, svikle na svakojake grubosti i otuda ne oseća potrebu da u komunikaciji sa korisnikom bude naročito prijatan.

## Crtanje kola

Organizacija programa EDIT je prilično neuobičajena za savremene grafičke pakete — koji se po pravilu zasnivaju na sistemu menija i sličica — ali i izuzetno efikasna. Upravljanje jednim grafičkim programom procenjuje se prema jednostavnosti s kojom crtač rukuje osnovnim grafičkim elementima. Zahvaljujući malom broju elemenata kojima manipuliše u programu *smARTWORK* je primenjeno izuzetno jednostavno rešenje. Programom se upravlja preko interne komandne linije, koja se otvara jednostavnim pritiskom na ENTER, a grafički elementi se crtaju funkcijskim i kursorskim tasterima, kojima se može ali i ne mora pridružiti i miš. Veličina i oblik stopice se određuju u komandnom režimu naredbom **pad**, a stopice se postavljaju i uklanjaju funkcijskim tasterima F3 i F4. Postavljanjem stopica može se nacrtati podnožje bilo koje elektronske komponente — od otpornika do integrisanih kola s proizvoljnim brojem nožica — ali bi to bio dosta zametan posao.

U programe za projektovanje i crtanje štampanih kola obično se ugrađuju biblioteke podnožja (foot prints library) koja se pozivaju jednostavnim navođenjem simbola (poput DIP40 ili slično). Autori *smARTWORK*-a primenili su i ovde originalno rešenje — program je opremljen posebnim naredbama za njihovo definisanje **sip** (single-inline package — jednoredna integrisana kola), **dip** (dual-inline package — dvoredna integrisana kola), **d** (dvopolne polarizovane komponente poput dioda i elektrolitskih kondenzatora), **r** (nepolarizovane komponente sa dva izvoda poput prekidača i otpornika) i **conn** (dvoredni konektori za trakasti kabl). Uz izvesne razlike u upotrebi, naredbe zahtevaju čitav niz parametara kojima se opisuju karakteristike podnožja — smer u odnosu na kursor (strane sveta **n**, **s**, **e** i **w**), ukupan broj nožica, rastojanje između redova nožica, rastojanje između samih nožica ili polaritet. Svakom podnožju se opciono može odrediti i ime (site — U1 za integrisano kolo ili C1 za kondenzator) i *smARTWORK* će tada numerisati svaku nožicu — nožicama u integrisanom kolu dodeljuje redne brojeve (U1 1, U1 2 ...), podnožjima definisanim komandom **d** a (anode) i k (katode), a komponentama sa dva izvoda a i b. Kod integrisanih kola i konektora stopicama se mogu dati i simbolička imena naredbom **term** (na primer, U1 a0, U1 a1 ... U2 a0, U2 a1 ... ako je u pitanju adresna magistrala na memorijskom čipu) ali je posao izuzetno zametan, a kod dvopolnih komponenti tipska imena se mogu menjati u okviru same naredbe. Ove pogodnosti olakšava crtanje (broj ili simbolička oznaka stopice na kojoj se nalazi kursor ispisuje se u statusnoj liniji) i bez nje se ne mogu ni zamisliti rad autorutera i automatske kontrole štampanog kola.

Ove naredbe pružaju crtaču punu slobodu da definiše podnožja različitih veličina ali uz poštovanje dosta strogih pravila — podnožje se, na žalost, može definisati samo na mreži najmanje gustine 2,54 mm (100 stotih delova inča) što isključuje izvestan broj savremenih komponenata koje imaju sitniji raster.

## Tipski parametri

*SmARTWORK* spada u izuzetno zatvorene programe. Možete ga ili prihvatiti onakvim kakav jeste ili — odbaciti. Korišćenjem opcija koje prethode programu prilikom njegovog pokretanja neki parametri se srećom ipak mogu promeniti. Postoji međutim i nešto elegantniji način — promene se mogu uneti u tipsku datoteku (datoteku) koju programi EDIT, DOT, PLOT i NET prilikom startovanja traže pod imenom WINTEK.DEF ili WINTEKDEF. U ovu datoteku se mogu uneti boje za svaku radnu površinu, tip video kartice, kataloško ime crteže na disku, tip skaliranja, veličine stopica, brzine komunikacije sa ploterom, smer crteže, debljina tankih linija, format štampe, širina papira, tip plotera, format u kome je zapisan spisak spojeva i veza. U najlakši se nalazi čist tekst poput **padsize=75r** (veličine stopice 0.075 inča, oblik kruga).

## smARTWORK

**Verzija**
1.4 (1 februar 1988.

**Paket**
Dve disketa sa sistemskim softverom, demo disketa i priručnik

**Hardversko okruženje**
Dva flopi diska ili flopi disk i tvrdi disk, 512 K radne memorije, DOS 2.00 do 3.30

**Ulaz**
tastatura, miš

**Izlaz**
CGA, EGA i VGA grafičke kartice sa odgovarajućim monohromatskim ili monitorima u boji

**Štampa**
Matrični štampači „epson" serije FX/RX/MX, IBM grafički štampač „proprinter". Ploteri „Houston Instrument" serije DMP-40, 41, 42, 51, 52 i 56 i „Hewlett Packard" 7470A, 7475A, 7550, 7570, 7580, 7585 ili 7586

**Transportivost**
Prihvata spiskove veza u formatu *HiWire*, *Schema* i *OrCAD*

**Proizvođač**
Wintek Corporation
1801 South Street
Lafayette, Indiana 47904 2993 USA
Telefon (317) 742-8428

**Cena**
Verzija bez autorutera 495 US $
Verzija sa autoruterom 895 US $

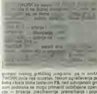
Naredbama, osim toga, nisu obuhvaćena neka izuzetno važna pakovanja kao što je **pin grid array** (mada se ono uz dosta ručnog rada može nacrtati), ivični i D konektori kao i čitav niz pasivnih komponenata čija se podnožja crtaju naredbama opšteg tipa, ali se ne unose automatski u crtež u belog sita. Grube mreže na kojoj radi i skroman izbor komponenata predstavljaju, bez sumnje, ključne slabosti programa *smARTWORK*.

Vodovi u *smARTWORK*-u se mogu povlačiti samo između postavljenih stopica ili u nastavku legalno započete linije. Kompletna organizacija programa podređena je upravo jednostavnom i efikasnom crtanju veza. Početak i kraj voda se označavaju tasterom F1, uspravni i vodoravni vodovi se povlače kursorskim tasterima, a kose linije tasterima **Home**, **End**, **PgUp** i **PgDn**. Povučeni vod se može podebljati tasterom **F5** i ponovo svesti na normalu tasterom **F6**. Poslednju povučenu vezu je moguće ponoviti neograničeni broj puta, što značajno ubrzava crtanje simetričnih vodova — povezivanje memorijskih čipova na nekoj procesorskoj ili memorijskoj ploči crtač obavlja izuzetno brzo i bez imalo napora. Vod se može crtati u proizvoljnom broju poteza korišćenjem F1 za označavanje prelomnih tačaka. Pri tom se izuzetno lako prelazi sa jedne na drugu stranu

Pored ručnog crtanja veza, u *smARTWORK* je ugrađena i pogodnost za samostalno povlačenje linija najkraćim putem između dveju označenih stopica. Iako se crtač štampanog kola neće baš uvek složiti sa estetikom na kojoj su zasnovana pravila po kojima program samostalno povlači veze, projektanti su spremni da zbog ove opcije oproste *smARTWORK*-u sve njegove grubosti, slabosti i ograničenja.

Pogrešno povučene veze u *smARTWORK* u se briše sa izuzetnom, možda i previše velikom lakoćom — dok taster BACKSPACE jede liniju tačku po tačku, taster F2 odjednom uklanja čitavu vezu između dve stopice bez obzira na njenu dužinu i složenost i crtač može lako da izgubi njen kraj ili početak. Da bi se obrisao samo deo linije, najpre se mora postaviti a potom i posebno obrisati graničnik (puna tačka). U program nije ugrađena ni naredba tipa UNDO koja bi omogućila crtaču da se predomisli i povrati pogrešno obrisanu liniju. Brisanje vodova srećom nije jedini izlaz iz problema na koje crtač nailazi prilikom projektovanja štampanog kola: da bi se napravilo mesto za još jedan vod, nije potrebno brisanje i premeštanje postojećih — naredba cleave stvara novi prostor potpuno automatski. Ona, međutim, nije dovoljna nadoknada za strogost programa prema korisnicima — *smARTWORK* zahteva od crtača štampanog kola da uvek zna šta radi i da bude potpuno koncentrisan na svoj posao.

Rad sa grupama podataka predstavlja važnu po-



godnost svakog grafičkog programa, pa ni *smARTWORK* ovde nije izuzetak. Nakon označavanja početka i kraja bloka tasterom F9, nad izdvojenom grupom podataka se mogu primeniti uobičajene operacije — brisanje, presikavanje, premeštanje i popunjavanje zatvorenih površina — ali samo u njihovom najelementarnijem obliku. Izdvojeni delovi crteža se može odložiti u privremenu biblioteku na disk i potom uzeti i preneti na neku drugu ploču. Ugradnja elemenata iz biblioteka, ako se tako uopšte može nazvati mehanički niz crteža na disku, predstavlja izuzetno zametan posao — *smARTWORK* ga automatski postavlja donjim levim uglom na poziciju kursora i ostavlja crtač može da ima tačnu predstavu o njegovoj veličini. Stoga se crteži prihvataju u prazan deo u radnom prostoru i tek potom naredbom MOVE postavljaju na svoje pravo mesto. Cinik bi rekao da je i to znatno lakše od ponovnog crtanja čitavih blokova. Sve jedno da su se programeri malo više trudili, crtači bi imali znatno manje ručnog rada.

## Automatsko oblikovanja

Greške u oblikovanju štampanog kola mogu da imaju visoku cenu. Izrada prototipske ploče nije ni laka, ni brza, ni jeftina, i otuda pouzdanost jedne tehnike projektovanja i mogućnost provere rezultata njenog rada predstavljaju važnije kriterijume od njene efikasnosti. U *smARTWORK* je ugrađen mehanizam koji — pod uslovom da mu se saopšti tačan spisak spojeva i veza (netlist) i da se prema tom spisku postave i označe komponente — obezbeđuje potpunu pouzdanost projektovanja i sigurnu kontrolu štampanog kola. Ovakvi spiskovi se pripremaju u posebnim programima za crtanje električnih shema. Na tržištu postoji desetak profesionalnih programa sa ovom namenom koji, treba li to reći, generišu isto toliko različitih formata. Program *smARTWORK* direktno prihvata format iz programa za crtanje shema *HiWire*, a pomoću programa za konverziju NETCVT.EXE i formate iz programa *OrCAD Future-Net* i *Schema*. Električna shema je ionako polazna tačka svakog ozbiljnog projektovanja — na njoj se greške najlakše uočavaju i ispravljaju — a spisak spojeva i veza koji ona generiše samo razlog više da se projektant štampanog kola upusti u njeno crtanje.

Spisak spojeva i veza predstavlja oslonac za dve krucijalne osobine *smARTWORK*-a — automatsku kontrolu tačnosti i automatsko crtanje štampanog kola. Autoruter se oslanja na složeni skup unutrašnjih pravila a njegov rad se može prekinuti u svakom trenutku da bi mu se pomoglo da povuče neku kritičnu vezu. Nakon prekida, program može da nastavi rad ili od početka ili sa mesta na kome je zaustavljen. Program na kraju posla izdaje izveštaj o broju povučenih veza, ali ne i o tome koje veze nije uspeo da povuče. Do ovog podatka se srećom lako dolazi posredno — primenom naredbe rlist koja u kombinaciji sa znakom „–“ izdaje spisak nepovučenih, a sa znakom „+“ spisak povučenih veza.

Autoruter kojim je opremljen program *smARTWORK* ne ume samostalno da prenosi veze sa jedne na drugu stranu i ne ume samostalno da menja raspored komponenata — ova opcija je u završnoj fazi razvoja — i taj nedostatak bitno smanjuje njegovu efikasnost. Procenat uspešnosti je međutim još uvek veoma visok — crtač može sa sigurnošću da računa da će program umesto njega povući između 80% i 90% veza. U demo primeru mikroprocesorske ploče koji se dobija uz program autoruter povlači 98% veza (315 od 320) za 12 minuta (XT sa V20 na 8 MHz) sa uključenim prikazom na ekranu. Iako radi prema strogim pravilima za dizajniranje štampanih kola, autoruter ponekad ima čudan smisao za estetiku — ako ima dovoljno prostora na ploči, više voli kosu liniju od prave — ali se sve to lako da ispraviti. Pri tom se vreme meri minutama — beznačajno u odnosu na časove koji su potrebni za ručno projektovanje na računaru ili dane ručnog crtanja klasičnom tehnikom.

Bez obzira na način projektovanja štampanog kola — ručnim radom, autoruterom ili nekom od mogućih kombinacija — kontrola tačnosti je brza, jednostavna i efikasna. Program *NETGEN* pretražuje crtež štampanog kola i sastavlja spisak ostvarenih spojeva i veza. Ovaj spisak se programom *NETCMP* poredi sa spiskom stvarnih veza. Izveštaj sa rezultatima kontrole i popisom grešaka se, u zavisnosti od odabrane opcije, šalje ili na ekran ili u posebnu datoteku.

## Štampanje crteža

Pojednostavljeni crtež štampanog kola na ekranu monitora svoj pravi oblik dobija tek kada se prenese na papir programima *DOT* ili *PLOT*. Program *DOT* pogoni matrične štampače u „epson“ i IBM standardu i pomoću njega se mogu raditi dva osnovne vrsta crteža — kontrolni crteži u razmeri 1:1 i 2:1 za proveru tačnosti i originalni crteži (camera ready artwork) u razmeri 2:1 koji se direktno koriste u produkciji štampanih kola. Prvu vrstu crteža *DOT* proizvodi izuzetno brzo, uz tehnički kvalitet koji je sasvim dovoljan da zadovolji njihovu svrhu. Kod druge vrste štampanja crtež prolazi kroz preciznu obradu i finu doradu — stopice prave i kose linije, prelomi na linijama i spojevi između stopica i linija dobijaju svoj konačni oblik i potrebnu punoću — i štampač obično ima pune ruke posla. Finoća crteža na matričnom štampaču je oduvek bila problematična i *DOT*, na žalost, tu ništa ne može da popravi — ivice stopica i linija su iskrzane onoliko koliko je to neophodno, ali se to u znatnoj meri gubi umanjenjem crteža na prirodnu razmeru prilikom njegovog prenošenja na grafički film. Program *PLOT* — proizvodi samo originalni crtež u razmeri 2:1 na papiru pausu i poliesterskom filmu koji predstavlja bolju osnovu za reprodukciju — i ploteri odazbeduju savršeni rezultat.

U fabričkoj proizvodnji dvostranog štampanog kola koristi se šest crteža. Program *smARTWORK* automatski crta tri, a programi *DOT* i *PLOT* iz njih izvlače svih šest: crtež donje i gornje strane, masku donje i gornje strane, crtež belog sita sa rasporedom komponenata i raspored stopica. Vrsta crteža zajedno sa još nekoliko obaveznih parametara izričito se moraju saopštiti programu pre početka crtanja, pri čemu se izbor može proizvoljno kretati između jednog i svih šest. Svaki crtež se crta na posebnom papiru, a *DOT* i *PLOT* samostalno odlučuju kako će on biti orijentisan — ako je previše velik da stane popreko na izabrani papir, biće nacrtan uzdužno ili, ako i to nije moguće, podeljen na nekoliko celina koje se ručno spajaju lepljenjem. Oko svakog crteža *DOT* crta okvir i u zaglavlju ispisuje najosnovnije podatke — broj stopica i približnu veličinu ploče u inčima.

Podatke neophodne za svoj rad — tip crteža, širina valjka, vrsta izlaznog uređaja, vrste crteža, debljina tankih linija, brzina serijskog prenosa, dimenzije papira, model plotera — programi *DOT* i *PLOT* prikupljaju kroz sistem upita i menija na početku rada. Iskusniji korisnici mogu ove podatke da saopšte i preko komandne linije prilikom poziva programa ili da ih ugrade u DEFAULT datoteku po imenu WINTEK.DEF ili WINTEK.DEF i time poštede sada uvek istih zamornih propitkivanja. Upravljanje programima *DOT* i *PLOT* je, uz poštovanje određenih hardverskih specifičnosti, gotovo u svemu istovetno.

## Sa obe na zemlji

„Wintek“ je, pre svega, kompanija za projektovanje i razvoj hardvera. *SmARTWORK* je nastao kao razvojni program na moćnoj mašini (PDP 11/44) za internu upotrebu i tek potom, na nagovor poslatilace fabrike, prenet na PC. Ovaj gotovo bi se moglo reći slučajan prodor ima i svoju na tako malu cenu — *smARTWORK* je program ne samo bez čvrsta nego i pomalo zastarele koncepcije: ograničena veličina i mogućnost projektovanja samo dvoslojnih štampanih kola, previše gruba mreža za crtanje projekata sa modernim procesorima u novim kućištima i konektorima sa nestandardnim razmakom između pinova, ograničen izbor dimenzija stopica i debljina vodova, stroga i tvrda komunikacija sa korisnikom.

Bez obzira na sve svoje manjkavosti *SmARTWORK* još uvek sa oba noga čvrsto stoji na zemlji — njegove su vrline veće i značajnije od njegovih mane: brzo, lako i efikasno crtanje, automatsko povlačenje pojedinačnih veza, interaktivno samostalno crtanje štampanog kola, jednostavna kontrola tačnosti i, konačno, izuzetno doteran, pouzdan i kvalitetan crtež. To je dovoljno da uđe u onaj najuži krug programa zbog kojih vredi kupiti PC.

*Jova Regasek*

## Tehničke karakteristike

Dimenzije štampe 10•16 inča
Broj slojeva 2
Raster 0.050 inča
Ugrađena podnožja DIP i SIP
Debljina linije 0.012, 0.015, 0.020 i 0.050 inča
Razmak između linija 0.050 (90°) i 0.05 (45°) inča
Veličina stopica 0.062 i 0.075 inča
Oblik stopice krug, kvadrat

## Od sheme do štampe

Proces modernog projektovanja i crtanja štampanog kola odvija se u nekoliko tipičnih faza:

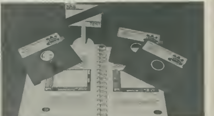
- crtanje električne sheme u programu *HiWire*, *OrCAD* ili *Schema*
- izdvajanje spiska spojeva i veza iz projektno i tehnički tačnog crteža
- konvertovanje spiska u *HiWire* format ako su korišćeni *OrCAD* ili *Schema*
- unošenje spiska u *smARTWORK* korišćenjem naredbe LOAD
- postavljanje i označavanje komponenata prema spisku spojeva i veza
- interaktivno korišćenje autorutera za izvlačenje većine veza — premeštanje komponenata u toku rada
- tehničke i estetske dorade štampanog kola — povlačenje veza koje autoruter nije povukao i korekcija veza čiji nam se izgled ne dopada
- izdvajanje spiska spojeva i veza iz štampanog kola programom *NETGEN*
- poređenje programom *NETCMP* stvarnog spiska sa referentnim spiskom iz programa *HiWire* ili *NETCVT*
- ispravke eventualnih grešaka i propusta
- štampanje crteža



750

# Žive slike

**Iako za PC ima nekoliko „crtačkih" programa, grafikom se na ovim računarima radi minimalno. Najčešće primene vezane su za poslovnu grafiku, ali se ovde javlja problem posebne vrste — dijafragmi, sheme i ilustracije se relativno lako konstruišu, ali je njihova prezentacija, na primer auditorijumu u toku nekog predavanja, gotovo nemoguća ili na krajnje amaterskom nivou. *GRASP* sistem za profesionalnu grafičku animaciju može veoma efikasno da reši mnoge probleme ove vrste**



Osnovu *GRASP* sistema čini jednostavan programski jezik opremljen sa oko šezdeset naredbi čiji je osnovni cilj da na što veći broj načina postave, uklone ili pomere bit-mapirane ilustracije. Jezik se bavi i ispisom teksta različitim pismima i veličinama, a raspoznaje i neke elementarne kontrolne strukture koje omogućavaju ponavljanje, grananje ili testiranje. Koncepcija je ipak, bliža na primer DOS bač datotekama nego pravim pogramskim jezicima, ali i to ima svojih prednosti: program je jednostavniji za učenje i omogućava „instant" rezultate bez poznavanja osnova programiranja.

## Klasičan editor

*GRASP* je namenjen PC/XT/AT računarima i isporučuje se na tri diskete. Instalisanje je veoma jednostavno — obezbeđeni su programi *FINSTALL* i *HINSTALL* koji program prenose na radne diskete ili tvrdi disk. Instalacija traje kratko — program smo za samo nekoliko minuta preneli na hard disk. Uz program se dobija uputstvo od oko 200 strana, koje je najvećim delom referentni priručnik u kome su vrlo pregledno zabeležene sintakse i načini upotrebe svake naredbe. Drugi deo priručnika od šezdesetak strana je posvećen programu *PICTOR* — grafičkom editoru.

*GRASP* se startuje navođenjem imena programa i na ekranu se dobija kontrolni program koji nudi neku vrstu integrisanog okruženja, na koje smo navikli iz Borlandovih kompajlera. Na vrh ekrana je postavljen meni koji omogućuje prelazak u tekst editor u kome se pišu GRASP programi, startovanje programa, snimanje na disk i pozivanje posebnog grafičkog editora o kome će kasnije biti reči.

Editor je po svemu klasičan programski editor koji razume podskup *WordStar* naredbi — sasvim dovoljno za pisanje programa od nekoliko stotina redova. Programi se snimaju i prihvataju u čistom ASCII formatu, tako da za pisanje možete konstiti i neki drugi editor na koji ste navikli. Osnovna prednost ovog editora je u ugrađenoj HELP opciji koja daje sintaksu svih GRASP naredbi.

Program se može startovati na više načina — opcijom RUN iz glavnog menija ili tasterom **F10** kojim se ujedno i zapisuje na disk. Taster **F9** je posebno pogodan u fazi testiranja programa jer omogućuje startovanje od linije na kojoj se nalazi kursor.

Kako izgleda *GRASP* program? Jedan jednostavan primer sa listinga 1 pružiće jasnu sliku.

```
VIDEO A
FLOAD ROMAN
PLOAD SLIKA1 1
PLOAD SLIKA2 2
TEXT 0,200,"ILUSTRACIJA BROJ 1",20
WAIT 100
PFADE 12,1
TEXT 0,200 "ILUSTRACIJA BROJ 2" 20
WAIT 100
PFADE 12,2
WAITKEY
EXIT
```

Listing 1: Jedan jednostavan *GRASP* program




**Iznenađenje iz paketa: Grafički editor *PICTOR***

Primeru gotovo da i nije potreban komentar. *GRASP* ima mogućnost odvajanja 16 bafera za slike, 128 bafera za isečke slike i bafer za font. U svaki od ovih prostora može se dinamički ubaciti odgovarajući sadržaj (PLOAD, CLOAD, FLOAD) i zatim nešto uraditi sa njim — postaviti ga na neko mesto na ekranu na jedan od 25 različitih načina naredbom PFADE, pomeriti po ekranu, dopisivati tekst ili docrtavati elementarne geometrijske forme — linije, pravougaonike, kružnice itd. Skup *GRASP* naredbi je tako izabran da maksimalno podrži sve pa i veoma specifične hardverske karakteristike najčešćih video adaptera.

## Ne samo nabrajanje

Ipak, *GRASP* program nije čisto nabrajanje naredbi. Dozvoljeno je i grananje naredbama za testiranje tastature ili miša (IFKEY, IFMOUSE), kao i skokovi klasičnim GOTO i GOSUB. Ponavljanja su obezbeđena MARK/LOOP kombinacijom, tako sa mogu pisati programi solidne složenosti. Bez preterivanja se može reći da se *GRASP*-om može kreirati neograničen broj veoma efektnih prezentacija na veoma profesionalan način. Vrhunac je, svakako, kreiranje programa sa menijima koji se aktiviraju mišem ili svetlosnom olovkom.

Osnovna naredba kojom počinje svaki GRASP program je VIDEO mode kojom se, u stvari, aktivira jedan od raspoloživih režima rada sledećih video adaptera: IMB CGA, IBM EGA, IBM VGA, HERCULES, HERCULES INCOLOR, AST COLORGRAPH-PLUS, PLANTRONICS i AST PREVIEW. Tako će na primer, VIDEO G, aktivirati režim EGA karte 640×350 sa 16 boja, VIDEO E aktivira EGA režim 640×350 sa dve boje, VIDEO H je namenjen „herkulesu" itd.

Svakako da se najefektnije prezentacije mogu dobiti uz pomoć adaptera sa visokom rezolucijom i velikim brojem boja, ali se u tom trenutku kao osnovni problem pojavljuje memorija — nekoliko slika je dovoljno da „pojede" čitav sistem, a dinamičko upisivanje znatno usporava čitavu prezentaciju.

## . I ne samo guranje miša

Jedan od načina da se kreira slika koja će biti korišćena u animaciji je i upotreba *PICTOR* programa koji se može pozvati iz *GRASP*-a, ili startovati kao nezavisan program. Ako poznajete PAINT-BRUSH program, onda znate sve i o *PICTOR*-u: izgled ekrana i izbor opcija je veoma sličan, iako je *PICTOR* nešto siromašniji. Kreiranje ozbiljnih slika u bit mapi je uvek zametan posao, pa ni *PICTOR* ne čini mnogo više od ostalih programa da ga olakša. Iako imamo relativno ograničena iskustva sa programima ovog tipa, *PICTOR* nam se odmah dopao — radi se o veoma odmereno koncipiranom programu kojim se lako rukuje, koji veoma brzo reaguje na naredbe i koji konačno priznaje činjenicu da grafički interfejs nije samo „besomučno" guranje miša po

stolu — tastatura ima podjednako važnu funkciju jer je nezamenjiva pri brzom aktiviranju režima i izvođenju naredbi.

*PICTOR* koristi PIC format zapisa, dok su fontovi zapisani u FONTRIX formatu. Uz program se dobija pet fontova (nezaobilazni ROMAN i OLD ENGLISH, BOLD, NORMAL i SPECIAL).

## Šou može da počne

Drugi način za kreiranje ilustracija je „hvatanje" ekrana drugih grafičkih programa. U okviru sistema se nalazi program rezidentnog tipa CAP. Program „uranja" kombinacijom ALT i desnog SHIFT tastera. Moguće je sačuvati sadržaj čitavog ekrana ili samo jednog dela koji se obeleži pravougaonikom. Program je veoma doteran — omogućava promenu kombinacije tastera koji ga aktiviraju da bi se izbegao sudar sa sličnim programima, forsiranje različitih video režima u trenutku preuzimanja slike, pa čak i snimanje u bejzik BSAVE formatu.

Ova dva programa daju posebnu vrednost čitavom sistemu. *GRASP* programi, slike, isečci itd. se uz pomoć GLIB-a mogu spakovati u jednu datoteku i bez ikakvih posebnih zahteva za pravo korišćenja, distribuirati zajedno sa *GRASPRT* programom. *GRASPRT* predstavlja „RUN-TIME" verziju interpretera koji izvodi programe iz biblioteka. Dovoljno je da se na disketi nađu *GRASPRT* i, na primer, biblioteka DEMO.GL i nakon
GRASPRT DEMO
šou može da počne!

*GLIB* dozvoljava nekoliko operacija sa bibliotekama — izbacivanje nekog elementa, listanje sadržaja u skraćenom i punom obliku, dodavanje programa i crteža, pa i raspakivanje biblioteke na sastavne datoteke.

## I drugi jezici

Srce *GRASP*-a, zapravo, čini zbirka potprograma koja proširuje set BIOS-ovih rutina u poznatoj funkciji INT10 i aktivna je samo dok je računar pod kontrolom samog programa. S obzirom da je jedna od naredbi *GRASP*-a i CALL, kojim se startuje bilo koji program sa diska, u principu je moguće koristiti i sledeću tehniku:

- startuje se GRASP
- izvede se naredba „CALL eksterni program" kojom se aktivira bilo koji EXE ili COM program, i

pošto je GRASP aktivan, program može putem INT10 direktno da izvodi GRASP naredbe.

Naredbe se prosleđuju putem pointera (ES:DX) na string u kome je u ASCII formi zapisana naredba.

GRASP po mnogo čemu predstavlja veoma doteran sistem koji dosledno sledi polaznu koncepciju. Utisak je, međutim, da se pri njenom izboru najviše vodilo računa o onom uskom krugu koji se bavi izradom programa za prezentaciju grafičkih mogućnosti video adaptera ili računara u celini, kakve smo navikli da vidimo po izlozima inostranih radnji, pa otuda verovatno i ono „za profesionalce" u nazivu programa. *GRASP* je verovatno dobro došao i profesionalcima u manjim TV studijima. Možda je ipak daleko šire polje primena *GRASP*-a vezano za neke manje ispitive namene. Tu pre svega mislimo na sasvim „amatersku" upotrebu računara kao sredstva za ilustraciju predavanja ili, kao edukativnog sredstva u nastavi sa nižim uzrastima. Za ovakve primene *GRASP* jeziku veoma nedostaje važan element — promenljive. Iako se one mogu nadomestiti kombinovanjem sa ostalim programskim jezicima, takva upotreba potpuno poništava njegove glavne prednosti — jednostavan i brz put do rezultata.

Program *GRASP* je izdao preduzimljivi Pol Mejs (Paul Mace) i može se nabaviti od izdavača po ceni od 55+10 dolara uključujući i poštanske troškove na adresu: *Paul Mace Software, 4000 Williamson Way, Ashland, Oregon 97520, USA.*

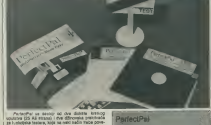
**Zoran Životić**

Komercijalni softver

Uslužni programi/PerfectPal

# Savršena svaštara

**Svaki novi programski paket opšte namene predstavlja izvanrednu priliku za manje softverske firme da ponude razne pomagale koje će u većoj ili manjoj meri ubrzati, odnosno olakšati rad sa glomaznim programom. Svega nekoliko meseci posle promocije tekst procesora *WordPerfect 5.0*, koji predstavlja školski primer čija složenost raste brže od njegove moći kalifornijske kompanije *PC Template* lansira zbirku preko 370 makro naredbi za *WordPerfect 5.0*, pod imenom *PerfectPal*.**




*PerfectPal* se sastoji od dve diskete, kratkog uputstva (25 A5 strana) i dva džinovska prekrivača za funkcijske tastere, koja na neki način treba povezati sa tastaturom ili držati u okolini kompjutera. Instalacija je sasvim jednostavna — pošto je katalog koji sadrži programski paket *WordPerfect* ionako stalno pretrpan, kreiraćemo njegov potkatalog (npr. \tekst\wp 5\pal), a onda, umećući svaku disketu u drajv A, kucati XCOPY A:PAL*.* C:\TEKST\WP5\PAL, obzirom na džinovski broj od preko 300 datoteka koje treba prepisati, čitav ća posao potrajati nekoliko minuta. Treba još startovati *WordPerfect*, ući u SETUP meni (Shift F1), izabrati opciju 7 *(Location of Auxiliary Files)* i saopštiti računaru da se makro naredbe i definicije tastatura kriju u katalogu *c:\tekst\wp5\pal*. Zatim, kada nam neka makro naredba zatreba, jednostavno pritiskamo *Alt* i odgovarajući taster *(Alt Q* je, na primer, *Quit)* ili kucamo *Alt F10* i ime makro naredbe. Ne može biti jednostavnije!

## Makro naredbe

Slika 1 prikazuje tastaturu čije su *Alt* funkcije predefinisane instalacijom *PerfectPal*-a. Vidimo da je svakom tasteru dodeljena po neka funkcija koja je uglavnom usko povezana sa odgovarajućim slovom: *Q* je *Quit*, *E Exit*, *R Retrieve*, *B Bold* i tome slično. Ideja autora *PerfectPal*-a bila je da pošteda korisnika od funkcijskih tastera — čim popamtite gde je šta na tastaturi, nećete morati da razmišljate o brojnim menijima i podmenijima *WordPerfect*-a. Pri upotrebi komercijalnih programa ne može se, jasno, zaobići neka varijanta zakona o održanju energije — 26 tastera nikako nije dovoljno da zameni nekoliko stotina stavki raznih menija! Autori *PerfectPal*-a su se, međutim, potrudili da na tastaturi postave za-

**PerfectPal**

*Da biste dobili svoj primerak PerfectPal-a* ... *dolara na adresu PC Template, P.O.* ... *Glendale, CA 91209, USA. U propratnom pismu treba specificirati verziju WordPerfect-a* ...

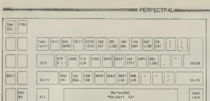
sta najpotrebnije komande, dok su korisnici složenijih opcija upućeni na imenovane makro naredbe.

Sastavni deo *WordPerfect*-a 5.0 je, kao što je poznato, svojevrstan programski jezik pomoću koga se definišu makro naredbe. Makro naredba jasno može da bude najobičnija sekvenca tastera (u toku pisanja ovoga teksta definisali smo, na primer, makro naredbe koje jednostavno ispisuju *WordPerfect* odnosno *PerfectPal* — tako štedimo dosta kucanja) ali i komplikovana struktura puna raznih INPUT i IF naredbi. Makro aktiviramo pritiskom na *Alt F10*, posle čega kucamo ime neke od naredbi sa slike 2 i pritiskamo ENTER. *PerfectPal* je konstruisan tako da naredbe kojima su potrebni neki parametri ispisuju odgovarajuće promptove u levom donjem uglu ekrana i očekuju da korisnik otkuca vrednost ili izabere stavku iz menija.

## Puno vrline...

Najinteresantnije makro naredbe bave se definisanjem oblika stranice i pisama.

Neke makro naredbe bave se isključivo oblikom teksta — naročito ističemo mogućnost da računar pretraži čitav tekst i podvuče, odnosno podebljа željenu riječ, frazu ili rečenicu. Nije za zanemarivanje ni mogućnost ukidanja isticanja teksta kroz čitav do-




Nova tastatura Slovnim tasterima su dodeljene najčešće naredbe

| LONGA | BAT | LORGB | BAT | LONGL | BAT | SHORTA | BAT | SHORTB | BAT | SHORTL | BAT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LET | WPK | SPANISH | WPK | STAT | WPK | ADVANCE | WPM | ALTA | WPM | ALTB | WPM |
| ALTC | WPM | ALTD | WPM | ALTE | WPM | ALTF | WPM | ALTG | WPM | ALTH | WPM |
| ALTI | WPM | ALTJ | WPM | ALTK | WPM | ALTL | WPM | ALTM | WPM | ALTN | WPM |
| ALTO | WPM | ALTP | WPM | ALTQ | WPM | ALTR | WPM | ALTS | WPM | ALTT | WPM |
| ALTU | WPM | ALTV | WPM | ALTW | WPM | ALTX | WPM | ALTY | WPM | ALTZ | WPM |
| APPEND | WPM | ASCII | WPM | AUTOGEN | WPM | AUTOREF | WPM | BACKUP | WPM | BALL | WPM |
| BEEP | WPM | BTN# | WPM | BINDING | WPM | BLANK | WPM | BLOCK | WPM | BLOCKFR | WPM |
| BOX | WPM | BTEXT | WPM | BUALL | WPM | BWORD | WPM | CLEAR | WPM | CODE | WPM |
| CDL | WPM | COL2 | WPM | COL2NEWS | WPM | COL3 | WPM | COL3NEWS | WPM | COL4 | WPM |
| COL4NEWS | WPM | COLDEF | WPM | COLMAKER | WPM | COLOR | WPM | COMMENT | WPM | COMMENTE | WPM |
| COMPOSE | WPM | CONDENSE | WPM | CONDPAGE | WPM | CONVERT | WPM | COPY | WPM | COPYCOL | WPM |
| COPYCOLT | WPM | COPYDOC | WPM | COPYLINE | WPM | COPYMERG | WPM | COPYM | WPM | COPYPARA | WPM |
| COPYSENT | WPM | COUNT | WPM | CPAGE | WPM | CTEXT | WPM | CURSOR | WPM | CW | WPM |
| DATE | WPM | DATEFORM | WPM | DATEFUNC | WPM | DIR | WPM | DISPLAY | WPM | DOCCOPPA | WPM |
| DOS | WPM | DRAFT | WPM | DUPLICAT | WPM | ELITE | WPM | EMPHASIZ | WPM | END | WPM |
| ENGLISH | WPM | ENVELOPE | WPM | EXLARGE | WPM | EXPANDED | WPM | FALL | WPM | FASTSAVE | WPM |
| FCART | WPM | FDEFAULT | WPM | FDOWN | WPM | FIELD | WPM | FINITIAL | WPM | FLUSHRTG | WPM |
| FN2REF | WPM | FN2TXT | WPM | FNORMAL | WPM | FOOTER | WPM | FORCE | WPM | FORM | WPM |
| FORMFEED | WPM | FRESET | WPM | FSIZE | WPM | GRAPH | WPM | GRAPHDEF | WPM | GRAPHQ | WPM |
| HALFLINE | WPM | HARD | WPM | HEADER | WPM | HYPHEN | WPM | I2TAB | WPM | TLR | WPM |
| INDEX | WPM | INDEXALL | WPM | INDEXDEF | WPM | INDEXGEN | WPM | INITIAL | WPM | ITALIC | WPM |
| JADJUST | WPM | KERNING | WPM | KEYMAP | WPM | KEYMAPLE | WPM | KEYMAPST | WPM | LABELGEN | WPM |
| LANDSCAP | WPM | LANGUAGE | WPM | LASTCUR | WPM | LAW | WPM | LEGAL | WPM | LEGALDEF | WPM |
| LEGALEDI | WPM | LETTER | WPM | LH | WPM | LINE | WPM | LINE# | WPM | LIST | WPM |
| LISTDEF | WPM | LISTGEN | WPM | LOCK | WPM | LOWERCAS | WPM | LPI | WPM | LPI6 | WPM |
| LPI8 | WPM | MADJUST | WPM | MARK | WPM | MASTERCO | WPM | MASTERDE | WPM | MASTEREX | WPM |
| MATH | WPM | MATHC | WPM | MATHDEF | WPM | MBOTTOM | WPM | MCOND | WPM | MDEFAULT | WPM |
| MEASURE | WPM | MEASURES | WPM | MELETE | WPM | MEMO | WPM | MERGE | WPM | MERGECOD | WPM |
| MERGEDEF | WPM | MONITOR | WPM | MOVE | WPM | MOVECOL | WPM | MOVECOLT | WPM | MOVELINE | WPM |
| MOVEM | WPM | MOVEPARA | WPM | MOVESENT | WPM | MPICA | WPM | MR | WPM | MRESET | WPM |
| MTOP | WPM | NLQ | WPM | NOTE | WPM | NOTEIT | WPM | NOWINDOW | WPM | OALIGN | WPM |
| ODEF | WPM | OLECAL | WPM | ORAKER | WPM | OUTLINE | WPM | OVERSTRT | WPM | PI | WPM |
| P2 | WPM | PADDRESS | WPM | PADJUST | WPM | PAGE | WPM | PAGE# | WPM | PAGE#NEW | WPM |
| PAGE#OFF | WPM | PAGE#POS | WPM | PAGE#TEX | WPM | PAGELENG | WPM | PAL | WPM | PALA | WPM |
| PAPER | WPM | PARA# | WPM | PARA2SEN | WPM | PBLOCK | WPM | PCOLOR | WPM | PCOMMAND | WPM |
| PCOPY | WPM | PDOC | WPM | PDOS | WPM | PGO | WPM | PINITIAL | WPM | PINSTALL | WPM |
| PITCH | WPM | PITCHADJ | WPM | PLII | WPM | PL13 | WPM | PL14 | WPM | PLABEL | WPM |
| PMENU | WPM | POPTIONP | WPM | POPTIONT | WPM | PPAGE | WPM | PPAGE# | WPM | PRETURN | WPM |
| PREVERSE | WPM | PREVIEW | WPM | PROPORTI | WPM | PROTECTB | WPM | PRUSH | WPM | PSTATUS | WPM |
| PTEST | WPM | RDOC | WPM | RECORD | WPM | REDLINE | WPM | REDSET | WPM | REF | WPM |
| REF# | WPM | REF2FN | WPM | REF2TXT | WPM | REFEDIT | WPM | REFOPTIO | WPM | REFORMAT | WPM |
| REFPLACE | WPM | REPEAT | WPM | RLOCK | WPM | RULER | WPM | S42 | WPM | SBLOCK | WPM |
| SCREEN | WPM | SD | WPM | SDOS | WPM | SEARCH | WPM | SEARCNT | WPM | SEARCHBA | WPM |
| SEARCHCO | WPM | SEARCHDE | WPM | SEESPACE | WPM | SENT2PAR | WPM | SETUP | WPM | SHADOW | WPM |
| SHORT | WPM | SHP | WPM | SMALLCAP | WPM | SMARK | WPM | SORT | WPM | SORTBLOC | WPM |
| SORTLINE | WPM | SORTMERG | WPM | SORTORDE | WPM | SORTPARA | WPM | SPACE | WPM | SPACER1 | WPM |
| SPACER2 | WPM | SPANISH | WPM | SPELL | WPM | SPELLRLO | WPM | SPELLDOC | WPM | SPELLPAG | WPM |
| SPELLWOR | WPM | SS | WPM | STANDOUT | WPM | STOR | WPM | STRIKEOU | WPM | STRIP | WPM |
| STYLE | WPM | STYLE1 | WPM | STYLE11 | WPM | STYLE13 | WPM | STYLE14 | WPM | SU | WPM |
| SUB | WPM | SUBDOC | WPM | SUMMARY | WPM | SUMMARYO | WPM | SUMMARYP | WPM | SUPER | WPM |
| SUPPRESS | WPM | TAB10 | WPM | TAB21 | WPM | TAB5 | WPM | TALIGN | WPM | TALIGNSE | WPM |
| TDEFAULT | WPM | THES | WPM | TIME | WPM | TIMEFUNC | WPM | TL | WPM | TOC | WPM |
| TOCDEF | WPM | TOCGEN | WPM | TOCLEGAL | WPM | TOP | WPM | TP | WPM | TRESET | WPM |
| TS | WPM | TW | WPM | TXT2FN | WPM | TXT2REF | WPM | TYPING | WPM | UALL | WPM |
| UDOUBLE | WPM | UPPERCAS | WPM | USINGLE | WPM | USTYLE | WPM | UTEXT | WPM | UWORD | WPM |
| WADJUST | WPM | WINDOW | WPM | WORD | WPM | XB | WPM | XBALL | WPM | XC | WPM |
| XCOL | WPM | XCOLALL | WPM | XCOLT | WPM | XCOMMENT | WPM | XF | WPM | XFALL | WPM |
| XFNALL | WPM | XHEADER | WPM | XHP | WPM | XIALL | WPM | XINDENT | WPM | XJALL | WPM |
| XJOFF | WPM | XJON | WPM | XKEYMAP | WPM | XLOCK | WPM | XM | WPM | XMALL | WPM |
| XMARK | WPM | XPAGE | WPM | XPAGE# | WPM | XPARA | WPM | XPLALL | WPM | XREFALL | WPM |
| XRSALL | WPM | XSEESPAC | WPM | XSENT | WPM | XSPACE | WPM | XT | WPM | XTAB | WPM |
| XTABALL | WPM | XTALL | WPM | XU | WPM | XUALL | WPM | XWINDOW | WPM | Z1 | WPM |
| Z2 | WPM | Z3 | WPM | | | | | | | | |

**Od svega po malo — za svakog po nešto; Spisak svih makro naredbi u paketu *PerfectPal***

kument (korisno kada probate novi drajver za štampač), prenošenje poruka i ključnih reči u druge datoteke, brisanje višestrukih blanko simbola između reči, pretvaranje blankova u tabulatore i tome slično. Autoru ovoga teksta su se međutim najviše dopale naredbe za rad sa kolonama — moram da priznam da me je od višekolonskog rada u *WordPerfect*-u uvek hvatala muka (kada šta aktivirati, gde šta definisati i tako u beskraj) dok se *PerfectPal*-om treba otkucati samo COL2 ili COL2 NEWS.

## ... i nekoliko mana

Pogled na sliku 2, sa druge strane, otkriva i suštinsku slabost sveštara kao što je *PerfectPal* — u gomili od preko 370 makro naredbi gotovo je nemoguće razdvojiti bitno od nebitnog i zapamtiti koja je naredba vredna pažnje a koja nije: pomenuta naredba COL2 je, na primer, veoma bitna, dok je naredba BLOCK krajnje nevažna jer ju je mnogo lakše zameniti prilikom na *Alt F4*. Da bi stvar bila još gora, pomoć koju pruža dokumentacija je veoma mala — veći deo uputstva posvećen je (ionako krajnje jednostavnoj) instalaciji *PerfectPal*-a, dok su naredbe pobrojane isključivo taksativno i bez ikakvih obaveštenja. Na dodatnom „čaršavu" naredbama su doduše dodata jednorečenična objašnjenja, ali ni ona ne mogu da se označe dovoljnim za normalnu upotrebu — svaku makro naredbu treba par puta probati kako bi se ustanovilo šta je koji parametar i šta se može a šta ne može urediti. Ovakvo isproba-



vanje jasno, odnosi mnogo vremena, što znači da se složenost *PerfectPal*-a pomalo približava složenosti samog *WordPerfect*-a ... možda će neka još manja firma uskoro ponuditi programski paket koji prilagođava *PerfectPal* prosečnom korisniku računara!

Druga mana *PerfectPal*-a je rastrošnost — 382 datoteke koje ga čine duge su ukupno 195 756 bajta ali na hard disku zauzimaju čitavih 837 632 bajta — gotovo koliko i pristojno instaliran *WordPerfect*! Radi se jasno o slabosti MS DOS-a koji alocira prostor u prevelikim klasterima, a ne o grešci programera firme *PC Template*, ali je to, u uslovima prepunog hard diska, prilično slaba uteha.

Programski paket *PerfectPal* je sve u svemu potencijalno veoma moćan i interesantan kako za početnike (ušteđeće mu brojne glavobolje) tako i za iskusnijeg korisnika koji će, analizirajući makro naredbe (treba pritisnuti *Alt F10*, otkucati ime i izabrati opciju 2 iz menija), naučiti mnogo toga o jednom od programskih jezika ugrađenih u *WordPerfect*. Šteta je jedino što spartanski kratka dokumentacija zahteva investiranje značajnog vremena u savladavanje svih tajni *PerfectPal*-a.

*Dejan Ristanović*

# Ne okreći se sine

**Sećate li se srećnih dana u kojima ste nas kritikovali što se bavimo Acornovim BBC-jem i „Elektronom", a ne važnim računarima kao što su „spektrum" i „komodor 64"? Ne može se osporiti da smo deo saveta uvažili: na vama je da kažete da li se dovoljno bavimo „spektrumima" i „komodorima" ali o BBC-ju u svakom slučaju više ne pišemo! Ili bar ne na velika vrata — pojava programskog paketa *BeebDOS* firme *MicroBoss* predstavlja dobru priliku da se, pišući od PC-ju, podsetimo starih, dobrih Acornovih kompjutera. Pokazuje se, međutim da život ide dalje. Ipak se ne treba osvrtati u nazad i gledati u stare računare.**

Iako IBM-ovi računari predstavljaju ideal raznih škola u svim krajevima sveta, britansko tržište je uvek imalo svoje specifičnosti. Većina engleskih škola je naime opremljena računarskim učionicama koje se sastoje isključivo od Acornovih računara: u centru je Master 512 ili (u novije vreme) Arhimed sa hard diskom a okolo je gomila modela B ili, ako je škola malo bogatija, Mastera; računari, jasno, komuniciraju preko mreže nazvane Econet. I pored uobičajene ljubavi prema sopstvenim proizvodima, Britanci su nedavno priznali da dobro obrazovani korisnik računara (o programerima da i ne govorimo) jednostavno mora da poznaje PC-ja i njegove najslavnije programske pakete, pa su sve brojnije škole koje su svoje kompjuterske učionice dopunile određenim brojem PC i AT klonova, najpopularniji je „amstrad PC" koji je uostalom britanski proizvod. Učionica u kojoj se nalaze računari raznih proizvođača koji su međusobno totalno nekompatibilni donosi brojne probleme sa prenosom podataka — zamislite da tekst unesen na „masteru" treba ponovo prekucati da bi se pojavio na PC-ju! PC kompatibilni računari se, uz sve to, relativno slabo uklapaju u Econet mrežu.

Pojedinci opremljeni BBC računarima se, kada pređu na PC, nalaze u još većim nevoljama — autor ovoga teksta je na primer na BBC-ju napisao nekih 10 megabajta stručnih napisa i knjiga, a onda obradu teksta (pre svega par meseci) prebacio na AT. Bilo bi veoma nezgodno kada bi stari tekstovi bili „zarobljeni" na BBC-jevim disketama i trakama i kada se njihovi delovi ne bi mogli koristiti u novim tekstovima i novim izdanjima! Kada je pre petnaestak meseci na naš sto došao prvi AT, sa BBC-jem je uspostavljena serijska (RS 232) veza koja nije baš slavno funkcionisala: maksimalna brzina od 1200 bauda i niska pouzdanost su prosto destimulisali ovakav prenos. Zatim smo nabavili 1770 disk interfejs za BBC i program *MS DOS Copy* (još ćemo ga pominjati u ovom tekstu) i sve je radilo kao san — BBC je snimao diskete koje su se bez problema učitavale na AT-u. I tako su stvari tekle dok, ljubaznošću firme *MicroBoss*, nismo došli do programskog paketa *BeebDOS*.

*BeebDOS* se izvršava na IBM PC-ju i zapravo predstavlja skup svih naredbi Acornovog DFS-a koje se mogu izvršiti na PC-ju: umesto da na BBC-ju otkucate *CAT, na PC-ju ćete otkucati BCAT, umesto *COPY na BBC-ju kucaćete BCOPY na PC-ju i tako dalje.

Instalacija *BeebDOS*-a je sasvim jednostavna: kreirate odgovarajući direktorijum (npr. */beebdos* ili */utility/bbc*) a zatim otkucate nešto poput *copy a:*.* c:/beebdos*. U određenom katalogu naći će se veliki broj relativno kratkih (po dvadesetak kilobajta) EXE programa koji zapravo predstavljaju komande simuliranog BBC-jevog operativnog sistema. Mnemonici su izabrani tako da asociraju na odgovarajuće BBC komande sa prefiksom B; ako vam se nešto ne dopada, odgovarajuću EXE datoteku možete slobodno preimenovati.

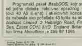
## Na popravnom ispitu

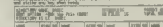
Koliko god se Acornu more odati priznanje na fleksibilno dizajniranom računaru, operativnom sistemu i bejzik interpretatoru, toliko mu se mora uputiti kritika zbog disk interfejsa. Acornov prvi disk interfejs za BBC B pojavio se krajem 1982. godine sa zakašnjenjem od skoro godinu dana. Bio je zasnovan na već tada prilično zastarelom Intelovom kontroleru 8271 koji je pre svega veoma skup (red veličine 70 funti) i prilično neracionalan: podržana je jedino standardna gustina zapisa *(single density)*, što znači da se na 80-tračnu disketu upisuje samo 400 kilobajta podataka, dok bi dvostruka gustina obezbedila gotovo dva puta više — 720 K. Acorn je zabrljao i kod softvera: DFS ROM ima samo 8 kilobajta i ne podržava nikakvu hijerarhijsku organizaciju disketa; na svaku se stranu može upisati svega 31 datoteka, programi za formatiranje i verifikovanje se isporučuju na uslužnoj disketi.



## BeebDOS

Programski paket *BeebDOS*, koji se sastoji od jedne diskete, relativno opsežnog uputstva (oko 40 A5 strana) i atraktivnih konca možete da nabavite ako pošaljete 43 funte na adresu *MicroBoss Limited, 3 Hebleigh Road, Frinton On Sea, Essex CO13 9HG, England*. Britanski telefon firme *MicroBoss* je 255 67 1095






Acornove propuste su začas ispravili nezavisni proizvođači, pa se na tržištu danas nalazi čitava gomila disk interfejsa i DFS ROM-ova za BBC B. Svaki od njih naravno podržava osnovni Acornov format, ali i mnogobrojne dodatke — kod nas naročito popularni Watford DFS uz komande za verifikaciju i formatiranje ugrađene u ROM podržava upis 62 datoteke na svaku stranu diskete, što je uglavnom dovoljno za normalan rad. Nevolja je jedino u tome što se diskete pri čijem je formatiranju korišćena opcija „62 datoteke" ne mogu potpuno učitati na BBC-ju koji je opremljen samo Acornovim DFS ROM-om.

Softverske promene ubrzo nisu bile dovoljne — Solidisk i Watford su tokom 1984. počeli da prodaju disk interfejse zasnovane na drugim kontrolerima. Watford je odabrao neposrednog naslednika Intelovog 8271 čija je oznaka 8272, dok se Solidisk mnogo ispravnije odlučio za Western Digital 1770. Tako su nastali razni DDFS *(Double Density Filing System)* interfejsi i ROM-ovi koji na standardne diskete upisuju 640 ili 720 kilobajta podataka. Ovako formatirane diskete su jasno potpuno neupotrebljive na BBC računarima koji su zasnovani na starom 8271, ali su Watfordovi i Solidiskovi programeri obezbedili mod kompatibilnosti koji omogućava učitavanje i snimanje standardnih disketa na „dvostruko gustim" interfejsima, emulacija je toliko dobra da može da „proguta" i zaštićene igre u standardnom Acornovom formatu.

Negde početkom 1986. godine Acorn je paralelno sa serijom „master" lansirao novi disk interfejs zasnovan na kontroleru 1770. Uz disk interfejs s

### Komande BeebDOS-a

Sintaksa većine komandi *BeebDOS*-a ima dva oblika: jedan odgovara BBC-jevom DFS-u a drugi ADFS-u. Eventualni argumenti se u bilo kom slučaju razdvajaju blankovima ili zarezima, dok tačka i zarez iza nekog argumenta nalažu računaru da za preostale argumente usvoji podrazumevane *(default)* vrednosti — baš kao da radimo sa nekim od Microsoftovih kompajlera! Razlika između velikih i malih slova nisu naročito dosledno izvedena, što znači da se u nekim komandama moraju koristiti isključivo velika slova, dok pri zadavanju imena datoteka o ovome ne treba voditi posebnog računa.

Sastavni deo paketa *BeebDOS* je program BCONV, neka vrsta siromašnog makro procesora koji treba da obezbedi konverziju između formata nekih BBC datoteka i odgovarajućih PC standarda. Opšti format naredbe kojom se konverzija izvršava je **BCONV <kontrolna datoteka>,<izvor>,<odredište>**. Kontrolna datoteka (standardna ekstenzija ALT) je zapravo najobičnija datoteka sa tekstom u kojoj u sukcesivnim redovima piše šta čime treba zameniti. Opšti format ove datoteke najbolje ćemo razumeti na primeru sa slike 2 koji vrši rudimentarnu konverziju VIEW datoteka sa tekstom u format razumljiv za razne PC-jeve tekst procesore. Kažemo *rudimentarnu konverziju* jer BBC na kraju svakog reda upisuje po jedan 0D bajt dok PC tekst procesori ovaj bajt upisuju samo na kraju pasusa; na BBC-ju se kraj pasusa prepoznaje po prvom znaku sledećeg pasusa. Obzirom da BCONV ne može da podrži ovako komplikovane zahteve, autor ovog teksta će i dalje koristiti svoj posebno napisani program na turbo paskalu za konverziju na relaciji VIEW — WordPerfect 4.2.

Na *BeebDOS* disketi data je i veoma značajna .ALT datoteka koja omogućava konverziju tokenizovanih programa na BBC bejziku u ASCII format.

isporučuje novi 1770 DFS ROM koji je po svemu sličan starom 8271 DFS-u: na disketu se upisuje 400 K podataka, na svakoj strani može da se nađe samo 31 datoteka i tako dalje; razlika je jedino u tome što je softver prilagođen novom kontroleru. Za nekih dvadesetak funti se, međutim, može nabaviti ADFS ROM koji predstavlja sasvim novi operativni sistem: na disketu se upisuje 640 K podataka, uvedeni su hijerarhijski organizovani katalozi i složeni atributi za zaštitu datoteka, omogućena fragmentacija tj. produžavanje jednom zatvorenih fajlova i tome slično. Samo je po sebi jasno da se ADFS disketa ne mogu koristiti na BBC računarima koji nisu opremljeni 1770 disk interfejsom i ADFS ROM-om. U računare iz serije „master" je, sa druge strane, ugrađen ADFS ali ne i stari 8271 interfejs, što znači da neke zaštićene igre umeju da prave probleme.

Autor ovoga teksta je krajem prošle godine nabavio 1770 disk interfejs i ADFS ROM koji su mu bili neophodni za finalizaciju jednog komercijalnog programa, ali nikada nije bio posebno zadovoljan čitavom kombinacijom. Naša je ocena da je ADFS, pre svega, očajno spor: brza komunikacija sa diskom koja je uvek bila zaštitni znak BBC-ja je prilično degradirana! Fragmentacija, osim toga, nije baš sjajno rešena, što znači da s vremena na vreme treba koristiti *COMPACT i to po nekoliko puta uzastopce (!?). Pri svakoj promeni disketa treba dalje kucati komande *UNMOUNT i *MOUNT što je veoma napogodno. Prebacivanje datoteka sa DFS-a na ADFS uz pomoć *bejzik* programa sa uslužne diskete je nezgodno toliko sporo da se na njega odlučujete samo kada vas natera neka velika nevolja. Rad sa ADFS-om, sve u svemu, kod nas nije stekao apsolutno nikakvu popularnost, a nismo baš ubeđeni da je u Engleskoj situacija bitno drugačija.

## Pucanj u prazno

Pošto smo shvatili da ADFS nije ništa posebno, reći ćemo da se *BeebDOS* snalazi isključivo sa ADFS disketama — standardne DOS diskete su podržane samo u kombinaciji BBC sa 40 staza XT, 80-tračne diskete snimljene uz pomoć DFS-a uopšte nisu podržane, dok se 40-tračne ne mogu učitavati na AT drajvovima od 1,2 megabajta. Obzirom da nam pri ruci nije bio ni XT ni BBC sa 40-tračnim disk jedinicama, nismo uspeli da isprobamo prenos podataka preko DFS disketa. Isprobali smo samo ADFS prenos i ustanovili da je fascinantno brz (tekst od 66165 bajtova se posle svega osam sekundi našao na hard disku) i prilično pouzdan. Jedini bag se javljao kada bismo nehotice otkucali BCAT 0 pri čemu je u drajvu A bila obična PC disketa: *BeebDOS* bi se tako propisno zbunio da više ne bi čitao ni jednu jedinu (ispravnu ili ne) BBC disketu dok ne bismo otkucali jedno obično DIR A sa PC disketom u drajvu A.

Zanimljivo pitanje glasi: da li postoji opravdanje tehničke prirode za nedostatak kompatibilnosti sa standardnim DFS disketama ili se radi o propustu firme *MicroBoss*? Iako precizan odgovor ne možemo da damo, verujemo da su tehničke mogućnosti postojale: ako 1770 ugrađen u BBC može da čita diskete snimane na 8271, pa čak i da učitava zaštićene programe, zašto nešto slično ne bi mogao da uradi i 795 kontroler ugrađen u PC? Već pomenuti program *MS DOS Copy* firme *InvNet* uz to na BBC-ju koji je opremljen 1770 disk kontrolerom savršeno čita i snima podatke u 40-tračnom PC formatu, čak i na 80-tračnim disk jedinicama što znači da kontroler ima sve potrebne mogućnosti, smatramo da je osnovna mana *BeebDOS*-a njegova koncepcija.

Da rezimiramo: pošto nabavite *BeebDOS* nećete moći da ubacite BBC disketu u PC drajv i prenesete datoteke: moraćete najpre da prenesete podatke sa DFS-a u ADFS format i to na samom BBC-ju (koji mora biti opremljen 1770 disk interfejsom i ADFS ROM-om) pa tek onda da ovako konvertovanu disketu pročitate na PC-ju odnosno ako vaš BBC ima 80-tračne disk jedinice AT-u. Ako je vaš BBC već snabdeven 1770 disk interfejsom, proći ćete mnogo bolje ako koristite program *MS DOS Copy* — *BeebDOS* je istina nešto brži od njega (*MS DOS Copy* je dobrim delom pisan na bejziku), ali će zbirno vreme koje se sastoji od prebacivanja sa DFS-a na ADFS i sa ADFS-a na PC biti bitno duže od vremena koje bi *MS DOS Copy* potrošio za direktno kopiranje DFS — PC. Sve u svemu, programski paket *BeebDOS* firme *MicroBoss* nam je, ako se izuzme zadovoljstvo što smo ponovo u prilici da pišemo o BBC-ju, doneo malo toga. *Dejan Ristanović*



Javni softver

Korisne rutine /PDOS/ Siniša Đureković

# Sinišin gambit

**Redovnim čitaocima „Računara" dobro je poznat fenomen takozvanog „softvera u javnom vlasništvu" *(public domain software)* — sve su češći veoma kvalitetni programski paketi koji su, voljom autora, prepušteni javnosti i koje, prema tome, svako može slobodno da kopira i distribuira. Što se domaće scene tiče, preplavljeni smo softverom koji tretiramo kao da je *public domain* (ah, ti pirati), ali pravog softvera u javnom vlasništvu praktično i nema, ili ga nije bilo — nedavno smo dobili prvu *public domain* disketu sa programom PDOS *Siniše Đurekovića*!**

PDOS disketa se sastoji od standardnog programa za arhiviranje PKXARC v3.5 i arhiviranih datoteka *ftimer.arc*, *errorh.arc* i *ptask.arc*. Pošto kreirate odgovarajući direktorijum na hard disku, otkucacete *pkxarc a ptask.arc c:/katalog* (*ptask.arc* zatim treba zameniti sa *errorh.arc* i *ftimer.arc* kako bi se izvršila dekompresija svih datoteka) i sačekati nekoliko trenutaka — na disku se pojavljuje niz izvršnih i izvornih programa i tekstualnih datoteka. Izuzetno će se kod programa *ptask.arc* pojaviti nove arhivirane datoteke *system.arc* i *examples.arc* koje treba posebno raspakovati primenom istog programa.

## Presretač grešaka

Pošto su programi raspakovani, možete da počnete da ih koristite. Autoru ovoga teksta najviše se dopao *errorh* — radi se o modulu za *Turbo Pascal 4.0* koji obezbeđuje kontrolu izvršnih aritmetičkih grešaka. Pri radu sa *Turbo Pascalom* nekima mnogo nedostaje bejzik konstrukcija ON ERROR GO TO... koja bi nas lišila neprekidnog ispitivanja argumenata — da li je delilac nula, da li rezultat veći od najvećeg celog odnosno racionalnog, da li je argument logaritamske funkcije manji od nule... Paskal program koji će koristiti usluge ovog „presretača *runtime* grešaka" treba započeti sa *uses ErrorH*, modul inicijalizovati sa *InstallErrorH*, a zatim, posle svake „sumnjive" aritmetičke instrukcije, koristiti funkciju *ArithError* koja vraća nulu ako je sve u redu ili neki od kodova grešaka 200—207. Drajver se deaktivira sa *UninstallErrorH*.

Program *ftimer* ubrzava sistemski tajmer 128 — u nekim je prilikama naime neophodno meriti vreme u jedinicama koje su preciznije od pedesetinki sekunde. Startovanjem *Turbo Pascal* programa koji počinje sa *Uses {$U Timer} Timer* i poziva proceduru *Speed(N:Byte)* tajmer se ubrzava $2^N$ puta. Za N=0 dobija se normalna brzina tajmera od 18,2 otkucaja u sekundi; ova neutralizacija je neophodna da bi se izvršavanje programa normalno okončalo.

## Paralelno procesiranje

Program *ptask* je najvažnija komponenta čitavog sistema. Čitaoci časopisa Byte su, verovatno, upoznati sa programskim paketom *ctask* (takođe *public domain*) koji omogućava simulaciju paralelnog procesiranja na C-u. *Ptask* je sličan paket koji će obradovati korisnike kod nas daleko popularnijeg *Turbo Pascala*. Ne treba, naravno, očekivati da će se ovako kratak program takmičiti sa raznim operativnim sistemima za 80386 — radi se o jednostavnim procedurama koje omogućavaju da se proizvoljno mnogo procesa vrti paralelno i to bez posebno rešenih prioriteta.

PTASK prilikom kreiranja novog procesa u *heap*-u otvara prostor za stek koji se prilikom „ubijanja" procesa oslobađa za druge procese. Pošto rutine *Turbo Pascala* za upravljanje memorijom nisu *re-entrant*, paralelno izvršavanje bi dovelo do nepredvidljivih stanja steka, pa je uvedena interna kritična zona *MemoryManager* koja dopušta samo jednom procesu da alocira ili dealocira memoriju. Ovakva koncepcija otežava korišćenje paskal procedura *new* i *dispose* i funkcija *MemAvail* i *MaxAvail*, ali obezbeđuje pravi pravcati multiprogramski rad koji je ilustrovan primerima *channels* (implementacija kanala), *hamming* (rešavanje Hammingovog problema), *matrix* (paralelno množenje matrica) i *fc2* (paralelno kopiranje datoteka).

Ulazak u avanturu zvanu paralelno procesiranje nije, na žalost, uvek bezbedan: u okviru dokumentacije pominju se „simpatični" uticaji višestrukih poziva nekih sistemskih procedura *Turbo Pascala* na FAT, direktorijum i druge osetljive zone — možda je bolje da pre startovanja instalirate Mejsovu vakcinu!

Nepotpun pregled programskog paketa uverio nas je da se radi o veoma zanimljivim rutinama koje će zaljubljenicima u paralelno procesiranje koji (još?) nisu nabavili razne transpjuterske pločice i prateće razvojne sisteme pružiti mnogo zadovoljstva. U autora se, najzad, može imati puno poverenje — ako nas sećanje dobro služi, *Siniša Đureković* je svojevremeno bio autor jednog od apsolutnih bestselera u našem Katalogu programa za džepne računare! Dokumentacija je relativno kompletna, ali i rascepkana u razne datoteke što znači da smo pripremajući ovaj prikaz ko zna koliko puta učitavali i brisali razne segmente teksta sa raznih mesta.

## Šta ko sme?

Obzirom da je ovo prva prilika da predstavimo neki domaći *public domain* program, red je da izložimo kodeks ponašanja u ovakvim situacijama. Program u javnom vlasništvu možete slobodno kopirati i distribuirati uz uslov da pri ovoj operaciji ne ostvarujete nikakvu zaradu — cena programa ne sme da pređe cenu medija na kom se isporučuje. Pravila takođe zahtevaju da paket kopirate u celini, uključujući tu i dokumentaciju, odnosno *procitaj.me* datoteku koja sadrži poruku autora. Ukoliko niko u vašoj blizini nema *ptask* koji biste prekopirali, možete se obratiti samom autoru na adresu *Siniša Đureković, Kikićeva 18, Zagreb*.

*Dejan Ristanović*

MCODER2

# Poslednji Mohikanac

**Program za kompajliranja bejzika MCODER2 nadživeo je sve svoje prethodnike, kao i kompajlera koji su se javili posle njega. Razloge za to treba tražiti delom u kvalitetu MCODERA2, a delom u manjkavostima ostalih programa.**

Od nekoliko bejzik-kompajlera za „spektrum" preživeo je samo jedan MCODER2. Njegov prethodnik, MCODER, zaboravljen je čim se pojavila nova, bolja verzija. SUPER C, IS, IS FP i drugi nikad nisu ni našli uporište među hakerima, jer je svaki od ovih programa imao po neki ozbiljan nedostatak koji je prevođenje činio neizvesnim i u krajnjoj liniji neupotrebljivim.

BLAST je svojevremeno bio prava senzacija, reklamiran je kao kompajler koji može da prevede sve reči Sinklerovog bejzika, bez izuzetka. Međutim, i on je ubrzo napušten, jer zauzima toliko memorije da korisniku ne ostaje prostora za iole ozbiljniji program. Ta mu dođe kao prepuna trpeza a gosti nemaju gde da sednu.

I tako je poslednji Mohikanac među kompajlerima ostao jer potvrđuje svoje dobre osobine:

a) Pouzdanost,
b) Veliku brzinu (40 do 100 puta je brži od bejzika)
c) Solidan fond reči
d) Jednostavno rukovanje
e) Tri režima rada.

MCODER2 ima, naravno, i odlike koje se mogu nazvati nedostacima. Ali na trebe biti cepidlaka — zašto gledati zube jedinom konju spoctnom da trči?

## Rukovanje

MCODER2 zauzima gornji deo memorije, od 56610 do 65367, to jest do bloka sa UDG znakovima. Posle je normalno odmah ispod UDG stek, pre učitavanja MCODERA2 treba spustiti RAMTOP. Određivanjem RAMTOPA ujedno se određuje adresa od koje će početi izvršni kod, takozvani kod. Zgodno, zar ne?

Ako se, na primer, RAMTOP spusti naredbom CLEAR 45000, bejzik i stek će biti ispod ove adrese, a kompajlirani program će se startovati sa **RANDOMIZE USR 45000.**

Na kraju kompajliranja MCODER2 izveštava o dužini izvršnog koda, tako da se blok sa kodom jednostavno snima na traku naredbom **SAVE „ime" CODE adresa, dužina.** Za učitavanje je dovoljno otkucati **LOAD ""CODE.** Naravno, pre učitavanja treba spustiti **RAMTOP.**

Prevedeni bejzik neće raditi ako u memoriji nije i sam kompajler. Neki korisnici zato snimaju na traku preveden zajedno s MCODEROM2 u jednom bloku. Kako se to praktično izvodi?

Recimo, program je preveden na 45000, a prema dobijenom izveštaju dug je 3500 bajtova, što znači da zauzima prostor do adrese 48500. Da ne bismo snimali na traku i prazno memoriju od 48500 do 56610, menjamo **RAMTOP** naredbom **CLEAR 55999,** ponovo kompajliramo i dobijamo prevod koji zauzima prostor od 56000 do 58500. Pošto je 65327−56000+1=9328, prevedeni bejzik i MCODER2 snimamo zajedno naredbom

**SAVE „ime"CODE 56000, 9328.**

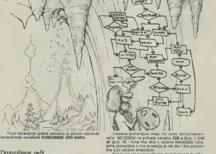
Ne, računar ume i da računa, a USR „a" ukazuje upravo na gornju adresu bloka plus 1 (65328), pa se možemo poslužiti varijantom koja nas oslobađa obaveze da utvrđujemo dužinu:

**SAVE „ime"CODE 56000, USR „a"−56000**

Samo kompajliranje je jednostavno do te mere da jednostavnije ne može biti. Treba samo narediti:

## Randomize USR 60000.

I nista više. **MCODER2** prevodi u dva prolaza. Prilikom prevođenja lista bejzik. Ukoliko je sve u redu, na kraju izveštava da je kompajliranje OK, koja je startna adresa i koja je dužina izvršnog koda. Ukoliko u bejziku naiđe na nedozvoljenu reč ili na grešku u sintaksi, prekinuće listanje, a kursor će postaviti upravo na mesto greške. Znači, dovoljno je da pritisnete CAPS i 1 da biste editovali liniju i potom ispravili grešku, a ne možete se dvoumiti oko toga šta je greška, gde je, jer kursor pouzdano ukazuje na nju.




Posle ispravljanja greške potrebno je ponovo startovati kompajliranje naredbom **RANDOMIZE USR 60000.**

## Dozvoljene reči

**MCODER2** radi samo sa celobrojnim vrednostima, što mu se računa u glavni nedostatak. Ali tako piše u uputstvu, a vešt programer će, koristeći funkciju INT, umeti da kompajlira program sa realnim brojevima. No to, kao i mnoge složene doskoke, pripada drugoj temi: naprednom korišćenju programa.

Celobrojne vrednosti su moguće od −32768 do 32767. Kompajler će, međutim, kao polazne brojeve u računanju prihvatiti sve brojeve do 65535. Na primer, sasvim korektno će izvršiti naredbu **PRINT PEEK 56000**.

S nekim izuzecima, kompajler dozvoljene reči shvata isto kao operativni sistem „spektruma". Zato još nešto o posebnostima MCODERA2.

| MCODER2 prevodi sledeće naredbe i funkcije | | |
|---|---|---|
| AT | PRINT | CODE „char" |
| BEEP | RANDOMIZE | CODE INKEY$ |
| BORDER | READ | IN |
| BRIGHT | REM | INKEY$ |
| NEXT | RESTORE | INT |
| OUT | RETURN | PEEK |
| OVER | STOP | POINT |
| PAPER | TAB | RND |
| PAUSE | ABS | SGN |
| PLOT | ATTR | SQR |
| POKE | SIN | USR |

Naredba **BEEP .5,10** neće biti prihvaćena, jer .5 nije celobrojna vrednost. Ali, umesto toga može se napisati **BEEP 1/2,10.** Uopšte, svaka neceIobrojna vrednost može se izraziti razlomkom (ili deljenjem, kako hoćete).

Indeksne promenljive mogu biti samo jednodimenzionalne. MCODER2 ne prihvata naredbu **DIM a (b,c)** ili **DIM a$ (b,c)**. Ali i tome ima leka u rukama Mandušića Vuka: jedna promenljiva s dve dimenzije je isto što i dve promenljive a po jednom dimenzijom.

Kad je već reč o promenljivima, da kažemo i to da njihova imena mogu biti ista kao u bejziku, da ih MCODER2 prepoznaje bez obzira na to da li su napisane malim ili velikim slovima i da stringovi ne mogu biti duži od 200 znakova.

Petlje **FOR - NEXT** mogu biti samo pozitivne. Nije dozvoljeno napisati, na primer, **FOR a=10 TO 1 STEP −1**. Ali ima i jedna povoljnost: pošto MCODER2 u svim petljama ne koristi ni svoj stek ni stek „spektrumovog" operativnog sistema, mogućan je korektan rad programa s uslovnim izlaskom pre okončanja petlje (iako se neko sada naježio zbog odstupanja od programerskog dogmatizma).

Iza **GO TO** i **GOSUB** mora se uvek napisati broj programskog reda. Nisu dozvoljene naredbe **GO TO** promenljiva ili **GOSUB** promenljiva.

**STOP** za MCODER2 ima sasvim drugo značenje. Ta je naredba za prekid kompajliranja. Drugim rečima, ostatak bejzika iza naredbe **STOP** neće biti preveden.

MCODER2 ignoriše **REM**, pa ovu reč možete koristiti na uobičajen način bez bojazni da će kompajlirani bejzik biti duži ili sporiji. Ali **REM** ima još jednu funkciju: da odredi jedan od tri režima rada. Sintaksa za određivanje režima je

**REM #broj**

Ova naredba se uvek piše u prvom programskom redu bejzika. Broj može biti od 0 do 2.

**REM #0** onemogućuje prekid programa pritiskom na CAPS i BREAK, osim u situaciji „scroll?" i kad se izvršava INPUT. U ovom režimu program se najbrže izvršava.

**REM #1** dozvoljava brejkovanje kao u bejziku. U ovom režimu izvršni kod je nešto duži i sporiji.

**REM #2** takođe dozvoljava brejkovanje. U ovom režimu, koji je spor i pogodan samo za debagiranje, u vrhu ekrana se prikazuje broj programskog reda koji se upravo izvršava.

Ukoliko režim rada nije određen, podrazumeva se **REM #1.** Proizvođač MCODERA2 je engleska firma P S S iz Koventrija.

*Žarko Vukosavljević*

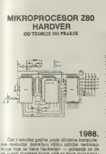
756

Dejan Ristanović

**Z. Darić i V. Jovanović**

**Mikroprocesor Z80 hardver (od teorije do prakse)**

**Izdanje autora, Otona Župančića 31, 1988. 120 strana, 20.000 dinara**




Čak i nekoliko godina posle oficijelne kompjuterske revolucije domaćem tržištu ozbiljno nedostaju knjige koje se bave hardverom — pokazuje se da na svakih dvadeset knjiga koje se bave programiranjem javi pet knjiga koje se bave upotrebom računara, pola knjige o hardveru! Knjiga sa pomalo rogobatnim naslovom „Mikroprocesor Z80 hardver od teorije do prakse" pretenduje da, bar kada se radi o popularnom Zilogovom mikroprocesoru, ispuni deo ovog informacijskog vakuuma.

Poput većine privatnih izdanja knjiga „Mikroprocesor Z80 hardver od teorije do prakse" ne izgleda naročito impresivno: mali format, bele korice, „lesetski" (začudo prilično atraktivan) prelom, dobar papir i skroman tiraž od 500 primeraka. Što se sadržaja tiče, knjiga se sastoji od devet delova i dvadesetak strana dodataka.

U okviru prvog dela opisuje se karakteristična arhitektura mikroprocesora sa tri magistrale (adresa, podaci i kontrola) i komunikacija mikroprocesora sa memorijom i periferijom. Ciljeve koncepcije se u okviru drugog dela konkretizuje na primeru mikroprocesora Z80, a zatim se u trećem delu opisuje koncepcija sistemskog oscilatora. Četvrti deo se, najzad, detaljno bavi komunikacijom mikroprocesora sa memorijom i periferijom i opisuje nekoliko karakterističnih konfiguracija mikroračunara.

Peti deo detaljno i kvalitetno opisuje prekidni sistem mikroprocesora Z80 i daje sugestiju za njegovu upotrebu, dok je u šestom i sedmom delu opisan rad pratećeg PIO čipa. Ostatak knjige predstavlja nekoliko interesantnih projekata: izlistura EPROM programator ... nema, naravno, projekta mikroprocesorske infuzione pumpa koji je autorima nedavno doneo Majsku nagradu grada Beograda za naučno-tehnička dostignuća. Dodatke čini rečnik korišćenih termina i niz tabela koja se odnosa na razne pripadnike familije Z80 i prateća 74LS kola. Čitav tekst je obilno ilustrovan kvalitetnim i funkcionalnim tehničkim crtežima.

Autori su se očito potrudili da relativno složenu materiju približe početnicima — iako knjiga ne podrazumeva praktično nikakvo predznanje, njeni pažljivi čitaoci će pošto razumeju i kreativno primene pomalo suvoparno izložene podatke, biti u stanju da projektuju jednostavnije sklopove (čak i mikroračunarski zasnovane na Z80. Ukoliko želite da slične projekte dovedete do kraja moraćete, jasno, da se opremite i softverskom literaturom slične prirode, pošto se, kako sami autori kažu u predgovoru, knjiga „Mikroprocesor Z80 hardver od teorije do prakse" bavi softverom samo kada je on neophodan za razumevanje rada mikroprocesora.

**Martina Radanović i Ivan Založnik**

**LOTUS 1—2—3 program za rad sa tabelama**

**ŠKO FORUM, 1988. 210 strana, 28.000 dinara**




U svetu neverovatno popularni programski paket *Lotus 1—2—3* postaje sve atraktivniji i u našim uslovima — brojni korisnici računara su očito primetili koliko se sve stvari može podvesti pod naoko jednostavan termin „tabelarni proračuni". Pri primeni programskog paketa *Lotus* neobično je, sa druge strane, zastupljen poznati princip 80:20 — osamdeset procenata aplikativaca koristi dvadeset procenata mogućnosti programa! Do većeg iskorišćenja se jasno dolazi prvenstveno proučavanjem literature koja je sve češće čak i na našem jeziku — dobar primer je knjiga „Lotus 1—2—3 program za rad sa tabelama" *Martine Radanović* i *Ivana Založnika*.

Knjiga „Lotus 1—2—3 program za rad sa tabelama" sastoji se od četiri dela, odnosno petnaest poglavlja. U prvom delu govori se o opštim stvarima kao što je istorija programskog paketa, priprema računara, početak rada, pristupni meni, ekran, i tabela. Veoma je korisno što se ne podrazumeva nikakvo posebno predznanje, što znači da će čak i korisnik koji nema nikakvo računarsko obrazovanje ili iskustvo biti u prilici da ispravno instalira program i, što je možda mnogo važnije, shvati šta će *Lotus* moći da učini za njega.

Drugi i po obimu najopširniji deo bavi se komandama paketa *Lotus 1—2—3*: rad sa tabelama, rad sa datotekama i ispis. Naročita pažnja je, naravno, posvećena poglavlju „Rad sa tabelama" u kome se detaljno opisuje preračunavanje tabela, kopiranje, premeštanje i brisanje ćelija i zaštita podataka.

Treći deo daje nekoliko konkretnih primera upotrebe paketa *Lotus 1—2—3*: izračunavanje cena proizvoda, izračunavanje kamata na vezana sredstva, vođenje skladišta i složeno poslovanje radne organizacije po kvartalima. Srećni smo što je konačno došlo vreme da o poslovnim programskim paketima progovore ljudi koji nisu ni matematičari ni elektroničari (autori knjige „Lotus 1—2—3 program za rad sa tabelama" su ekonomisti) i koji će pokazati da se *Lotus* koristi za vođenje poslova, a ne za neku statistiku radi statistike!

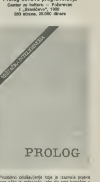
Poslednji deo zapravo predstavlja dodatak sa tabelarnim prikazom komandi i menija programskog paketa *Lotus 1—2—3* — pošto savladate ostatak knjige, dodatak će poslužiti kao zanimljiv referativni priručnik.

Knjiga „Lotus 1—2—3 program za rad sa tabelama" pisana je veoma stručno i nudi više nego detaljne informacije o materiji kojom se bavi. Pomalo suvoparno izlaganje uz preteranu upotrebu pasiva unekoliko je kompenzovano zaista atraktivnim prelomom (laserski štampači pokazuju šta mogu!), kvalitetnim papirom i dopadljivom tehničkom opremom. Sve u svemu, knjiga koju ni jedan potencijalni ili aktuelni korisnik *Lotusa* ne bi trebalo da se liši.

**Zoran Bjelica i Aleksandar Vavić**

**Prolog osnove programiranja**

**Centar za kulturu — Požarevac i „Braničevo", 1988 250 strana, 25.000 dinara**




Prvobitno oduševljenje koje je izazvala pojava prologa očito je splasnulo, tako da smo konačno u prilici da ovaj jezik takozvane veštačke inteligencije hladnije proučimo, zaključujući šta se sa njim može a šta ne može postići. Domaće literature koja bi nam pri tome pomogla bilo je i ranije (pomenimo samo priloge jednog od vrhunskih svetskih stručnjaka na ovom području profesora *I. Bratka*) ali je uvek interesantno videti šta o jednoj više nego kontroverznoj oblasti misli nova autorska ekipa. I tako smo pregledali knjigu „Prolog osnove programiranja" *Zorana Bjelice* i *Aleksandra Vavića*.

Knjiga „Prolog osnove programiranja" zapravo predstavlja prilično klasičan udžbenik prologa usmeren prema grupi korisnika koji bi želeli da naprave prve programerske korake na ovom jeziku. Knjiga se sastoji od osam poglavlja, priloga i spiska korišćene literature.

Prvi deo je, kao i obično, uvodni i bavi se karakteristikama i osnovnim pravilima prologa. Ta su pravila sistematizovana u drugom delu koji se bavi sintaksom, konstantama i promenljivima. Za sledeće elemente prologa zaduženi su treći (strukture), četvrti (liste) i peti deo (aritmetika), dok se u šestom delu detaljno opisuje za prolog neobično važni mehanizam povratnog pretraživanja *(backtracking)*. Posled-

nje dva poglavlja posvećena su često mašinski zavisnim komponentama prologa kao što su ugrađeni predikati i rad sa dibagerom. Prilog sadrži niz programa pisanih na prologu i to počevši od igre „život-inje" i rešavanja kvadratne jednačine pa do miniekspertnog sistema koji predstavlja sastavni deo Borlandovog programskog paketa *Turbo Prolog*.

Mora se reći da knjiga „Prolog osnove programiranja" ne donosi ništa što bi bilo posebno novo ili posebno originalno — redosled izlaganja je baš kao i većina primera preuzet iz stranih „bukvara prologa", što, uostalom, i sami autori spominju u predgovoru. Pomalo je, uz to, neprijatno što je dodat deo priloga preuzet u originalu tj. sa engleskim komentarima i engleskim promptovima — verujemo da za razumevanje knjige „Prolog osnove programiranja" nije potrebno preveliko računarsko predznanje ali da se bez solidnog poznavanja engleskog jezika jednostavno ne može!

Što se tehničke strane tiče, knjiga „Prolog osnove programiranja" predstavlja antipropagandu laserskih štampača — sa priličnom pouzdanošću tvrdimo da je reprofotografija izbegnuta štampanjem na pausu ili na foliji i da se dotični paus ili dotične folije nisu baš proslavili kvalitetom. Čitava knjiga je bleda od većine reprinta, a ponegde se javljaju i neprijatna data ostrva. Kratki primeri uz tekst složeni su *bold* fontom koji odudara od ostatka teksta kako „težinom" tako i dimenzijama, sve u svemu, nije nešto na čemu bi vam se pogled predugo zadržao!

Maksimalno afirmativni pogled autora knjige „Prolog osnove programiranja" na programski jezik o kome govore učinio je priručnik neinteresantnim za potencijalne čitaoce koji su zainteresovani za fenomenološke i komparativne karakteristike prologa i, uopšte, veštačke inteligencije. Knjigu zbog toga prvenstveno preporučujemo programerima koji bi želeli da se bez mnogo muke upoznaju sa raznim varijantama prologa za IBM PC i napišu jednostavnije programe koji će rešavati popularne probleme takozvane „veštačke inteligencije".

**Sybil P. Parker**

**Englesko-srpskohrvatski terminološki rečnik (računari — elektronika)**

**Građevinska knjiga, Beograd, 1988. 710 strana, 49.600 dinara**




Razvoj svake oblasti tehnike prati i razvoj odgovarajuće stručne terminologije. Kada se radi o oblasti dinamičnoj kao što je računarstvo, terminologija se naminovno stihijski razvija — reči nastaju i nestaju a značenje im se menjaju tako brzo da ih ljudi iz drugih struka teško prate. Vremenom su se, međutim, pojavljivali terminološki rečnici među kojima se, bar kada govorimo o računarstvu, posebno izdvajaju McGraw-Hill-ova izdanja. „Građevinska knjiga" je uložila značajan trud da ovaj u inostranstvu već klasičan rečnik prilagodi domaćim korisnicima.

Englesko izdanje (*McGraw-Hill Dictionary of Computers*, autor *Sybil P. Parker*) koje je poslužilo kao osnova za ovaj rečnik objavljeno je ne baš sasvim nedavne 1985. godine, mada je materijal štampan i ranije pod naslovom *McGraw Hill Dictionary of Scientific and Technical Terms*. Obzirom da su ravnopravan element rečnika brojni tehnički izrazi koji nisu vezani isključivo za računarsku tehniku, redaktori prevoda su (rekli bismo sa punim pravom) odlučili da nazivu dodaju odrednicu **elektronika**.

Složen posao prevođenja rečnika uspešno je obavio *Zoran Kovačević* u saradnji sa stručnim redaktorom *dr Spasojem Tešićem*, profesorom beogradskog Elektrotehničkog fakulteta. Za razne uže oblasti tehnike konsultovani su merodavni stručnjaci (uglavnom profesori ETF-a), što, na sreću, ne znači da se pojedini segmenti prevoda koncepcijski razlikuju.

Svaka stavka u rečniku sastoji se od engleskog naziva ispisanog masnim slovima, kratkog prevoda na srpskohrvatski jezik i, u sledećim redovima, opisa koji se obično sastoji od jedne ili dve rečenice. Reči su, jasno, sortirane po abecedi, pri čemu su principi sortiranja detaljno opisani na prvim stranicama rečnika — ostalo je neshvatljivo zašto autori sličnih publikacija ne nalaze za shodno da opišu kako su tretirali blankove, tačke, kose crte i ostale simbole u raznim imenima. Poslednjih stotinak strana knjige zauzima srpskohrvatsko-engleski rečnik, tj. prevod raznih domaćih izraza na engleski jezik.

Prevođenje pojedinih stručnih termina na srpskohrvatski jezik je veoma čest povod za razne polemike na stranicama (kompjuterske i ostale) štampe. Pošto verujemo da će izdanje ovog obima i ovog kvaliteta ostvariti značajan uticaj na dalji razvoj terminologije, pokušaćemo ukratko da opišemo pristup koji su autori primenili. Kroz čitav rečnik je dosledno poštovana želja da se što manje odstupa od originala, što znači da je priznato postojanje odomaćenih reči bafer, butstrapovanje, bas, stek, blocking oscilator, mod, set itd, svakoj od reči je, jasno, pridodato objašnjenje i poneki alternativni oblik (bafer, na primer, može da bude i prihvatna memorija, bas sabirnica i tome slično) ali je prilagođenje strane reči našem jeziku bez sumnje tu. Zanimljivo je, uz to, da strana imena nisu pisana fonetski (pominje se na primer Shockley-jeva diode, Hittorf princip, Rieke-ijev dijagram i tome slično) što je prilično retka pojava u istočnim krajevima Jugoslavije. Obzirom da se sasvim sličnih konvencija pridržava dobar deo saradnika „Galaksije" i „Računara", terminološka koncepcija terminološkog rečnika nam se izrazito dopala; ne sumnjamo, međutim, da će je mnogi čuvari čistote (i siromaštva) našeg lepog jezika dočekati „na nož".

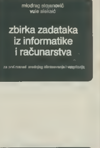
Pomalo paradoksalno za jednu knjigu od preko 700 strana, glavna zamerka „Terminološkom rečniku" mora da bude nedovoljan broj termina — uz sve ga nekoliko „štih proba" otkrili smo da nedostaju čak i tako važne reči kao što je *bold*, *laser printer*, *prolog*, *multisync*, *DOS*, *desktop publishing*, *spreadsheet* itd — u rečniku ćete, umesto ovih za čitaoce kompjuterskih časopisa svakodnevnih reči, naći brojne termine iz elektronike od kojih neki imaju samo istorijski značaj. Kompjuteri su, dakle, osnovna odrednica terminološkog rečnika, ali su personalni računari i njihove novije primene očito žrtvovani; možda uzroke treba tražiti i u tome što je originalni rečnik izdat još 1984. godine.

Bez obzira na ovo relativno siromaštvo termina, „Englesko-srpskohrvatski terminološki rečnik" predstavlja izuzetno značajnu publikaciju prema kojoj svi ranije izdati kompjuterski rečnici i leksikoni liče na igračkice. Radi se o knjizi koju treba da nabavi svako ko želi da se, proučavajući stranu literaturu, upozna sa računarskom tehnikom i elektronikom, dok će za nas koji pišemo o računarima „Terminološki rečnik" biti idealan saveznik. Ostaje samo da se nadamo da će nam ova izvanredna publikacija jednom biti ponuđena i u nekoj mašinski čitljivoj formi.

**Miodrag Stojanović i Vule Aleksić**

**Zbirka zadataka iz informatike i računarstva**

**Građevinska knjiga, Beograd 1988. 147 strana, 7.900 dinara**




Zbirka zadataka iz informatike i računarstva za prvi razred srednjih škola koja se pojavila početkom ove školske godine, obradovala je mnoge nastavnike i učenike iz prostog razloga što pre nje na našem jeziku nije postojala nijedna druga primerena znanju i mogućnostima prosečnih učenika ovog uzrasta. Miodrag Stojanović i Vule Aleksić držali su se data programa za predmet Osnovi tehnike i proizvodnje posvećenog informatici i računarstvu.

Autori su se trudili da organizacijom materijala, terminologijom i konvencijama za grafički prikaz algoritama slede udžbenik za ovaj predmet. Zadaci su podeljeni u dvanaest celina što bi trebalo da odgovara podeli fonda od 48 časova vežbi na blokove koji se realizuju kao jedna celina. U većini celina uz rešenja, ponuđeni su i zadaci za vežbu. Rešeni zadaci su uz objašnjenje i listing programa, propraćeni algoritmom i test primerima. Istina, bolje bi bilo da su rešenja data nezavisno od realizacije bejzika za jedan konkretan računar. Uz to bi i pored toga što se može diskutovati o kvalitetu ponuđenih rešenja, zbirka je ostavila bolji utisak, da su programi profesionalnije oblikovani. Ovo su, doduše, visoki zahtevi za rešenja pisana u bejziku, ali se mora imati u vidu da se u starijim razredima lakše radi sa učenicima koji ništa ne znaju o programiranju nego sa onima kod kojih treba ispravljati loše programerske navike. No, bez obzira na ove zamerke Zbirka zadataka iz informatike i računarstva koju je izdala „Građevinska knjiga" može korisno poslužiti u nastavi predmeta Osnove tehnike i proizvodnje jer nudi zadatke koje mogu pratiti učenici bez specijalnih predznanja iz matematike. Ova činjenica, uz ostale kvalitete daje knjizi prednost u odnosu na sve druge zbirke zadataka iz programiranja koje se trenutno mogu nabaviti u našim knjižarama, jer se može primeniti u nastavi osnova informatike u bilo kojoj srednjoj školi. Kako zasada ne postoji priručnik primeren potrebama informatičkog obrazovanja u osnovnim školama, zbirka Stojanovića i Vuletića trenutno predstavlja najprihvatljiviji materijal i za realizaciju izborne nastave iz računarstva u osnovnim školama.

(N.S.)

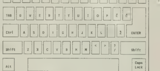
YU KEYBOARD SYS

# Reprogramiranje tastature

**Ugrađivanje YU slova u ROM neka grafičke kartice je rešenje polovine problema; druga polovina zahteva da ta slova rasporedite po tastaturi tako da se domaci tekstovi što jednostavnije unose i menjaju. Korisnici MS DOS-a 3.10 i 3.20 nisu imali mnogo problema sa definisanjem tastature (pogledajte „Računare 25"), ali prelazak na DOS 3.30 izaziva pravu glavobolju. Evo, dakle, u prošlim „Računarima" obećanog recepta za prilagođavanje datoteke KEYBOARD.SYS YU standardima.**



Slika 1 Američke verzije AT tastature



Slika 2 Tastature sa predefinisanim tasterima za YU latinicu

Pre nego što se zabavimo tehničkim problemima treba da odlučimo kako će tastature uopšte izgledati. Novi dizajn jasno zavisi od rasporeda dirki na vašoj tastaturi — širom Jugoslavije sreću se razne američke, nemačke, francuske, italijanske i ostale PC i AT verijante na kojima su, i pored približno jednakog izgleda, tasteri bitno drugačije raspoređeni. Kao osnovu za ovu operaciju uzeli smo dizajn originalne američke AT tastature sa 84 dirke čiji je deo (bez funkcijskih i numeričkih tastera) prikazan na slici 1.

## Dizajniranje...

Gde smestiti YU slova? Neophodno je, pre svega, pronaći pet tastera kod kojih je i osnovna i šiftovana funkcija malo bitna ili raspoloživa (bilo bi vrlo neprijatno da se, na primer, malo č skriva na tasteru | a veliko Č na tasteru @). Pažljivij pregled tastature dokazuje da takvi tasteri ne postoje. Iako bi se žrtvovale srednje i velike zagrade (dva tastera), obrnuta kosa crta i vertikalne crta (jedan taster), obrnuti apostrof i tilda (jedan taster), ali petog tastera jednostavno nema. Ostaje, dakle, da se odustane od slova Đ ili da se smisli neko međurešenje. To međurešenje je, u nakoj varijanti, dodeljivanje petog YU slova tasteru na kome su nacrtani apostrof i navodnik (počnite polako da spuštate pogled prema slici 2) koji, u svakom slučaju, predstavljaju znake bez kojih se ne može pisati tekst. U gornjem redu tastera se međutim pojavljuju neki znaci bez kojih se svakako može „majmun", „taraba", „and". Ideja se sama nameće: ovim tasterima dodelićemo neke simbole koje smo prethodno uništili čime se stvara zbrka (navodnik je, na primer, nacrtan na jednom tasteru a dobija se pritiskom na drugi) ali i rešava problem. Verujemo da ćete posle samo nekoliko dana pisanja uspeti da popamtite gde je šta čak i bez neozbeznih nalepnica na tasterima.

Pošto su tasteri na kojima će se intervenisati izabrani treba rasporediti znake. Odlučili smo da slova Ć, Č, Ž i Š smestimo u desni ugao tastature tako da budu grupisane — baš kao na većini pisaćih mašina. Đ je otputovalo u gornji levi ugao, dok su navodnik i apostrof smešteni iznad dvojke odnosno

```
Ins:00040  59 55 ..                                        .. ..  .   [YU-----  ----    .]
Ins:00110  .. ..  .....  .    . ..               59 55 .. ..          [-------  .    YU  ]
Ins:02240  .  .. .. AA .  20  . 30     A9    2F .. 3B .. AC           [---.-  =-  /  ,   ]
Ins:02250  .  AB .. A6 .                         .. .. .. ..          [-.-.-----       ..]
Ins:02260  .  .  80          5C    -5E    27  . 2A .. 28 ..           [  .---\-  -'-*-(-]
Ins:022C0  29 .. 5F .. 20 .. AE . -3A    AF .  B2        .. .         [)  _  +-  -       ]
Ins:022D0  3C .. 3E .. 3F .                              .. .         [< > ?-----  -   -]
Ins:02350  .  .. .. B1 ..      .                          .           [   ---  -----     ]
Ins:04C70  .. .  ..                   59 55 20   . . .  .             [--- -    -YU ----]
Ins:04CE0  . .. .. .. .. .. .. ..-.   59 55 20 73 65 74 20            [  ---- ---  --    ]
Ins:04CF0  62 79 20 44 2E 52 69 73-74 61 6E 6F 76 69 63 20            [--- --  -   --- -]
Ins:04D00  28 43 29 31 39 38 38 20-28 32 39 2F 34 2C 32 31            [--- -  -- .      ]
Ins:04D10  2F 38 29 20 .  .. .. ..-.. .. .. .. .. .. .. ..            [--  --    ---     ]
```

Slika 3 Podaci koje treba upisati u KEYBOARD, SYS da bi tasterima sa slike 2 bila dodeljena YU slova

sedmice — tamo se, na kraju krajeva, i nalaze na mnogim drugim tastaturama, na primer BBC jevoj. Sa slike 2 vidi se i da smo „tarabu" iznad trojke zamenili obrnutom kosom crtom koja se, istini za volju, u romanima i pripovetkama veoma retko srece ali koje je neophodne za zadavanje imana datoteka i direktorijume.

## ... i prilagođavanje tastature

Pošto je dizajnerski posao završen, prelazimo na programiranje. Sastavni deo DOS-a 3.10 i 3.20 bile su datoteka KEYBxx koje su u stvari sadržale rezidentne programe za definisanje tastature: učitate KEYBGR i prilagodili ste DOS nemačkoj tastaturi. Prepravka ovih drajvera je, kao što smo videli u „Računarima 25", krajnje jednostavna: pri kraju datoteke je serija bajtove koji opisuju koji taster daje koje slovo. Moramo, međutim, da kažemo da ni struktura starih KETBxx datoteka nije baš do kraja proučena — nije nam, na primer, poznato da je neko uspeo da reši problem računara čije se radne frekvencije menja sa Ctrl + odnosno Ctrl – (radi se o plusu i minusu sa numeričke tastature) i koji po učitavanju KEYBGR drajvera, odbijaju da izvrše ovu često neophodnu operaciju.

Bilo kako bilo, KEYBxx programi DOS-a 3.20 jednostavno ne rade na DOS-u 3.30 — po učitavanju se javlja jedino poruka *Incorrect DOS version*. Iz iskustva nam je poznato da je ova poruka uglavnom zaštitne prirode — EXE2BIN sa DOS-a 3.20 se, na primer, može ispraviti tako da preskoči početnu proveru i onda savršeno koristiti na DOS-u 3.30. Iako nismo pokušali da na sličan način „prilagodimo" KEYBxx drajvere, verujemo da se vredi potruditi oko propisnog rešenja koje bi odgovaralo DOS-u 3.30: nikad se, pre svega, ne može tvrditi da će svi programi koji će nam ubuduće doći do ruku biti raspoloženi za saradnju sa nekim „krpljenim" drajverom za tastaturu a, osim toga, DOS 3.30 nudi neke lepe novitete — zar nije zgodno da se sa Alt Ctrl F1 prelazi na američku a sa Alt Ctrl F2 jugoslovensku tastaturu i to usred bilo kog tekst procesora ili baze podataka?

DOS 3.30 je uveo novi sistem komunikacije sa periferijom koji se zasniva na kodnim stranicama i SYS datotekama. Učitavanje KEYBGR drajvera se, na primer, zamenjuje sa KEYB GR,KEYBOARD.SYS gde je KEYB eksterna naredba DOS-

-a GR naziv tastature e KEYBOARD SYS datoteka sa definicijama *svih* tastatura koje MS DOS podržava. Dimenzije ove datoteke (19766 bajtova) upozoravaju da analiza neće biti baš jednostavna

Pošto se naredba popul KEYB GR KEYBOARD SYS izvrši u memoriji ostaje rezidentni programčić KEYB (zauzima 5 5 K) koji veoma liči na bivši KEYBGR i u kome je prema tome vrlo lako pronaći i promeniti kompletnu definiciju tastature. Mogao bi se dakle napisati program koji pretražuje memoriju pronalazi KEYB i vrši direktnu izmenu uvodeći tako YU slova na tastaturu. Odlučili smo se ipak da potražimo pravo rešenje koje bi se zasnivalo na modifikaciji datoteke KEYBOARD SYS — zašto da svako aktiviranje YU tastature zahteva izvršavanje dva programa?

## Prepreka po sluhu

Početak datoteke KEYBOARD SYS sadrži niz ukazatelje na ukazatelje (!?) na pojedine tastature dva slova koja označavaju skraćenicu dva bajta ukazatelje dva slova koja označavaju skraćenicu i tako dalje. Na mestu na koja ukazuje ovakav pointer ponavlja se oznaka tastature, a zatim i blok podataka nepoznatog značenja obzirom da se dvobajtni ukazatelj na zonu u koju je smeštena sama definicija tastature nalazi na samom početku ovoga bloka uspeli smo da ga lociramo. Trebalo je još izabrati tastaturu koju bismo pretvorili u jugoslovensku što se pokazalo kao prilično složen zadatak. Većina tastatura naime podržava nekakve dvoznake i akcente što znači da se na primer kombinovanjem tastera A i E može dobiti znak AE ili da se kombinovanjem slova O sa raznim tasterima dobija O sa dve tačke, O sa crticom i tome slično

Moramo da priznamo da nam način na koji je rešeno ovo kombinovanje nije bio jasan pa smo se opredelili za italijansku tastaturu koja, u skladu sa uputstvom za PC DOS 3 30 ne koristi nikakve specijalne efekte ovoga tipa. Preciznu metodologiju rešavanja problema ne možemo da svedemo na reči — ofset definicije italijanske tastature je &2200 što znači da počevši od ove adrese treba tražiti grupe ASCII kodova koji podsećaju na raspored nekih tastera, zamenjivati ih probati, probati i probati. Ono što možemo da ponudimo jeste slika 3 na kojoj su dati podaci koje treba upisati u KEYBOARD SYS da bi tasterima sa slike 2 bila dodeljena YU slova raspoređenim u skladu *WordPerfect-u*) izmene možete da izvedete uz pomoć programa *Norton Utilities* pri čemu je osnova kopija datoteka KEYBOARD.SYS koja se isporučuje uz PC DOS ili MS DOS (razlika između PC DOS-a i MS DOS-a je jedino u tome što su na kraju PC DOS datoteka upisana imena ljudi koji su je kreirali dok je kraj MS DOS datoteka prazan)

Pošto unesete izmene i animirate datoteku pod imenom KEYBOARD YU, izvršite KEYB YU, KEYBOARD YU i pritiskajte sve tastere sa slike 2 proveravajući da li je svako slovo na svom mestu.

Ostalo je još da napis završimo jednim pitanjem pošto je na ovaj način instaliran KEYBOARD YU pritiscima na Alt Ctrl i neki taster se ponekad dobija njegova originalna funkcija — Alt Ctrl ć će, na primer proizvesti srednju zagradu. Slična kombinacija ne pali uvek Alt Ctrl š na primer ne proizvodi obrnutu kosu crtu. Da li je nekom od naših čitalaca uspelo da shvati ovu anomaliju i da definiciju tastatura dobere do kraja?

*Dejan Ristanović*





TOS /„atan"/ nedokumentovane opcije

# Bube u mišu

**Svaki operativni sistem, bez obzira koliko je dobro napisan, mora da ima nedokumentovane opcije i poneki bag, i TOS tu nije izuzetak. Ovaj članak pored poznatih, iznosi i neka bagova koji su do sada bili nepoznati korisnicima ST-a.**

„Atarijev" priručnik (User Manual) govori novom korisniku kako se otvara ili pomera prozor, ali nigde ne pominje da se direktorijum prozora može obnoviti pritiskom na taster ESC. Umesto da svaki put kada menjate disketu zatvarate prozor sa direktorijumom pa da ga po menjanju diskete ponovo otvarate, dovoljno je da zamenite disketu i pritisnete ESCape taster. Direktorijum diska će se obnoviti u istom prozoru

Isto tako, većina korisnika ST a nije otkrila svrhu desnog tastera na mišu — evo prilike da ga po koji put kliknete pri kopiranju fajlova morali prvo kliknuti prozor iz kojeg želite da kopirate da bi postao aktivan, pa tek onda vršiti selekciju fajlova i kopiranje. Umesto toga, dovoljno je da držite desni taster miša pritisnut dok levim birate fajlove — tako što ih kliknete. Zatim fajlove možete odvući do drugog diska ili ih baciti u đubre. Ne zaboravite da za vreme selekcije fajlova morate držati desni taster pritisnut.

## Rutina puna bogova

Možda će vam se desiti da prilikom inicijalizacije računar ne prihvati DESKTOP INF kako ste ga snimili na disk. Nemojte se plašiti — nisu u pitanju bad sektori na disku već jedna greška operativnog sistema da se sa DESKTOP INF fajlom ponekad mogu snimiti i totalno nepotrebni karakteri. Takođe ako radite sa RS 232 ne brinite ako stvari krenu naopako. Rutina koja kontroliše RS 232 je puna bagova (RTS/CTS i brzina od 50 i 75 bauda ne rade — lako ih verovatno niko i ne koristi). Zbog još jednog baga u AES-u nemoguće je odrediti path ABC..D jer sistem krahira. Pri pisanju ili čitanju sa diska sistem ne prijavljuje CRC grešku pri određenim uslovima i postoji mnoštvo nedokumentovanih sistemskih varijabli

Ako vam se kojim slučajem učini da miš ili tastatura otkazuju poslušnost — ne brinite. Isto se meni desilo pre nedelju dana i taman kada sam bio spreman da odvojim 130 DEM za novog miša rešenje je došlo samo od sebe. Naime svaki put kada sam probao da kliknem na bilo koju opciju u meniju ili *dialog box*-u kompjuter uopšte nije reagovao na signale sa miša. U panici da mi je kompjuter otkazao jer čak ni ALTERNATE + INSERT koji bi trebao da emulira levi taster miša sa tastature nije radio posegao sam za *mouse port* istrgao kabl iz konektora (da bih proverio da nije kojim slučajem prekinut kontakt negde u kablu miša). I ponovo probao ALTERNATE + INSERT i ovog puta je savršeno radio!

## Miš protiv treperenja

Tek kada sam ponovo probao da uključim miša u računar shvatio sam da u stvari miš nije ni bio izvučen, nego da je to bio džojstik na kome je uključen AUTO-FIRE. Zato, ako radite na računaru a imate džojstik u portu 2 pazite da vam nije uključen AUTO-FIRE. U slučaju da vam tastatura ili miš i dalje ne rade dobro moraćete da proverite čip za tastaturu. Primetio sam još jedan problem u vezi miša. Ponekad kada pritisnete dugme na mišu i pustite ga kompjuter iz nekog razloga i dalje veruje da je taster pritisnut. Ovo ćete videti najbolje ako postavite pointer na prazan deo desktop-a i držite levi taster pritisnut. Pojaviće se trepćući pointer i mali kvadrat *(box outline)*. Bez pomeranja miša otpustite dugme. U većini slučajeva pointer će prestati da trepери ali ako ovo probate više puta pointer će nastaviti da trepće iako dugme nije pritisnuto. Svako pomeranje miša će zaustaviti treperenje

Ovaj bag je nemoguće ispraviti, jer miša i tastaturu kontroliše IKBD čip *(Intelligent KeyBoard Device)* pa miš džojstik i tastature nisu pod direktnom kontrolom 68000 i ne postoje memorijske lokacije sa kojih se ova greška može prekinuti. Povremeno ćete videti varijacije ovog baga. Na primer kada kliknete na opciju iz menija a program ne reaguje na vaš izbor dok ne pomerite miša ili ako dvaput kliknete na program a on se ne učitava sve dok ne pomerite miša. Ovo je najlakše proveriti iz GFA BASIC-a postavljanjem miša u gornji desni ugao i pritiskanjem tastera SHIFT + F10. Program neće startovati dok ne sklonite pointer iz gornjeg desnog ugla. Možete probati i da kliknete dvaput na prazan desktop (van prozora) ali da zadržite levi taster pritisnut. Sa pritisnutim dugmetom pomerite pointer preko direktorijuma. Prvi program koji pointer dodirne će se startovati

## Vreme za zabavu

Takođe bi trebalo da pazite da nikada ne pokušavate da otvorite fajl iz ITEM SELECTOR a ako u drajvu nema diskete. Dobićete poruku „DRIVE A IS NOT RESPONDING " pa šta? No probajte da kliknete „RETRY" opciju a da pri tome ne stavite disk u A, a zatim kliknite na „CANCEL". Koliko bombi ste dobili? Opasnost isto postoji pri otvaranju fajlova iz bilo kog programa koji koristi ITEM SELECTOR. Kada vam se ITEM SELECTOR pojavi pritisnite UP ARROW (strelica nagore) da popnete kursor na DIRECTORY i otkucajte _ (SHIFT i —). Koliko ste bombi sad dobili? Ovi bagovi nisu u rutini ITEM SELECTOR-a nego u AES-u kod inputa u *dialog box*-ovima pa preporučujem da pazite na *underline* ( ) karakter dok koristite bilo koji *dialog box*

Ako ste nekim slučajem radili u ST bejziku (baš vas žalim što nemate GFA ili OMICRON bejzik) i otkucali X=18 9 sigurno ste dobili sledeću poruku FUNCTION NOT YET DONE SYSTEM ERROR #%N PLEASE RESTART. Nisam znao da je 18 9 tako težak broj. Najzabavnije tek dolazi! Otkucajte X=37 8 i pritisnite RETURN. Na trenutak će vam se pojaviti ista poruka dve bombe i vratićete se na desktop! Ako mislite da je već ovo previše pomerite miša.

Vlasnici „amiga" koji bi na ovo hteli da se nasmeju, mogli bi probati da u toku učitavanja ili snimanja programa izvade disk iz drajva

Nadamo se da će ovaj članak biti od koristi barem nekima od vlasnika „atarija" a ostalima koji nisu te sreće da poseduju ST poručujem da je „atari" i pored ovih bagova trenutno jedan od najmoćnijih personalaca.

*Dalibor Lanik*

# Editori teksta

**U četvrtom nastavku naše serije tekstova u VAX-ovom operativnom sistemu zvanom VMS bavimo se standardnim editorom teksta, koji se koristi kako za pisanje programa tako i za kreiranje i modifikovanje raznih datoteka koje obezbeđuju normalan rad sistema. Premda nije koncipiran kao tekst procesor opšte namene, EDT može koristiti i za pisanje kraćih tekstova i njihovo prilagođavanje nekom pravom tekst procesoru, kao što su *RUNOFF* ili *Tex*.**

EDT prirodno nije jedini pa čak ni najbolji editor koje je na raspolaganju korisnicime DEC-ovih računara iz serije VAX — razni *Language Sensitive* i drugi specijalizovani editori u pojedinim slučajevima predstavljaju daleko bolje rešenje. EDT se ipak isporučuje uz sam VMS, što znači da će ga u manjoj ili većoj meri koristiti svako ko je u kontaktu sa VAX-om. Činjenice da se EDT deli besplatno (tj. da je uračunat u cenu VMS-a) ovoga puta ne znači da se radi o slabom programu — EDT je izvanredan programerski editor koji je poslužio kao inspiracija za mnogobrojne slične programe za razne personalne računare i veće sisteme. Više puta se, uz to, pokazalo da osoblje računskih centara koji rade pod juniksom slavi dan zamene vi-a juniks verzijom EDT-a kao državni praznik!

## Iz linijskog u ekranski

EDT se startuje kucanjem komande **EDIT IME.TIP**, gde je sa *ime.tip* označeno ime tekstualne datoteke čiji sadržaj treba unositi odnosno modifikovati. VAX će najpre proveriti da li datoteka sa zadatim imenom postoji, a onda će, ako je odgovor na to pitanje potvrdan, kreirati novu verziju datoteke i unete staru verziju u specijalni bafer za editovanje. Pošto ispravljanje programa bude završeno, na disku će se naći kako nova tako i sve prethodne verzije datoteke *ime.tip* — prepunjenje diska sprečavamo povremenim kucanjem komande **PURGE** koja čuva samo najnovije verzije datoteke u radnom direktorijumu. Ukoliko, sa druge strane, VAX ustanovi da datoteka *ime.tip* ne postoji, biće kreirana njena prva verzija (*ime.tip 1*), ispisana odgovarajuća poruka i aktiviran editor — pretpostavlja se da ćemo sadržaj datoteke tek unositi.

EDT radi u dva modaliteta: posle **EDIT IME.TIP**, nalazimo se u linijskom editoru koji obezbeđuje kucanje komandi za obradu teksta: **DELETE 10** će, na primer, obrisati prvih deset linija programa. Za praktičan rad je, međutim, linijski editor krajnje neodgovarajući — treba otkucati **CHANGE** (biće dovoljno i jednostavno C, pošto se naredbe editora, poput naredbi VMS-a, mogu skraćivati) i kompletan će ekran biti posvećen tekstu koji ispravljamo. Ukoliko se po startovanju ekranskog moda pojave neke čudne poruke, pokušajte da napustite editor (komanda je **QUIT**) i u saradnji sa operaterom podesite parametre vašeg terminala — sva je prilika da on ne emulira VT100.

Rad sa ekranskim editorom je prilično očigledan za svakog iole iskusnog korisnika računara — uzastopnim pritiscima na tastere sa strelicama pomeramo kurzor po ekranu (kurzor će se, u stvari, stalno nalaziti negde oko sredine ekrana, što znači da će njegovo pomeranje biti realizovano kao pomeranje (skrolovanje teksta) i umećemo odnosno brišemo tekst. EDT se stalno nalazi u modu umetanja, što znači da će slovo koje otkucamo biti smešteno na poziciju kurzora dok će se ostatak linije pomeriti udesno. Iako dobar deo programera u praktičnom radu koristi uglavnom mod umetanja, smatramo da je nedostatak moda prekucavanja (*overtype*) osnovni i potpuno neshvatljivi nedostatak EDT-a.

Obzirom da je EDT programerski editor, osnovna jedinica sa kojom radimo je programski red. Red

| GOLD | HELP | FNDNEXT<br>FIND | DEL L<br>UND L |
|---|---|---|---|
| PAGE<br>COMMAND | SECT<br>FILL | APPEND<br>REPLACE | DEL W<br>UND W |
| ADVANCE<br>BOTTOM | BACKUP<br>TOP | CUT<br>PASTE | DEL C<br>UND C |
| WORD<br>CASE | EOL<br>DEL EOL | CHAR<br>SPECINS | ENTER<br>SUBS |
| LINE<br>OPEN LINE | | SELECT<br>RESET | |

**Zlatna tastatura: Uobičajeni raspored komandi za upravljanje editorom teksta prema takozvanom GOLD standardu**

počinje od levog ugla ekrana i prostire se do (nevidljivog) kôda <CR>. Kôd <CR> se dobija pritiskom na **RETURN**, što znači da pritiskom na ovaj taster dok se kurzor nalazi u sredini nekog reda zapravo cepamo tu liniju na dva dela; slično tome, pozicioniranjem kurzora na sam početak rada i pritiskom na taster <**DELETE**> uklanjamo <CR> karakter koji odvaja tekuću liniju od prethodne i tako faktički nadovezujemo redove jedan na drugi. Nekoliko proba će vam, čak i ako nikada niste radili sa nekim sličnim editorom, razjasniti ovu u suštini sasvim jednostavnu konvenciju.

U osnovni kurs EDT-a spada i upoznavanje naredbi za njegovo napuštanje. Obezbeđena je, pre svega, naredba **EXIT** kojom napuštamo editor uvažavajući sve izmene unesene u tekst. Naredbom **QUIT**, sa druge strane, napuštamo editor uz ignorisanje svih izmena koja su u toku seanse izvršene — ako je datoteka *ime.tip* već postojala, na disku se zadržava jedino njena prethodna verzija a ako datoteke nije bilo, sve što smo eventualno otkucali se uništava. Obzirom na destruktivnost, naredba **QUIT** se ne može skratiti na Q, QU ili QUI — treba otkucati čitavu reč.

## Nekoliko pikanterija

Treba, rekosmo, otkucati čitavu reč. Ali kako — reč koju otkucamo biće ugrađena u tekst koji obrađujemo, a ne izvršena kao komanda! Pre kucanja bilo koje komande treba preći u linijski editor za šta je zadužena kombinacija tastera **Ctrl Z**: kada se pojavi prompt linijskog editora u obliku zvezdice kucamo **EXIT** odnosno **QUIT** i pritiskamo **RETURN**. Ukoliko smo se greškom našli u linijskom editoru nema, jasno, nikakve prepreke da ponovo otkucamo **CHANGE** i tako aktiviramo ekranski mod.

Postoji još jedna pikanterija u vezi sa napuštanjem EDT-a: žurnal datoteka. Tekst koji obrađujemo se, što se nas tiče, nalazi u memoriji, dok se prethodna datoteka nalazi na disku (zahvaljujući principima virtuelne memorije, tekst koji obrađujemo se zapravo naizmenično šeta između memorije i diska, ali je to u ovom trenutku krajnje nebitno). Na disk se, međutim, upisuju i svi podaci o tasterima koje smo u okviru seanse sa editorom pritisnuli i tako nastaje datoteka čiji je pun naziv IME.JOU 1. Ukoliko primenom komanda **EXIT** ili **QUIT**, regularno napustimo editor, žurnal datoteka će biti automatski uništena. Ukoliko, međutim, editor bude prekinut neregularno (padne sistem, operator prekine naš proces, pritisnemo **Ctrl Y** i slično), žurnal datoteka se čuva, što znači da ćemo sa EDIT/RECOVER IME.TIP uspeti da vratimo datoteku u stanje u kome je bila nešto pre neregularnog prekida (biće, ukoliko je sistem opterećen, izgubljeno svega nekoliko poslednjih pritisnutih tastera). Čak i bez konfliktne situacije vredi pogledati „crtani film" koji EDIT/RECOVER proizvodi — započnite ispravljanje neke datoteke, a onda posle izvesnog broja operacija i izvesnog vremena pritisnite **Ctrl Y**, otkucajte EDIT/RECOVER IME.TIP, zavalite se u stolicu i posmatrajte!

## Zlatna tastatura

Iako su osnovne mogućnosti EDT-a koje smo upravo opisali dovoljne za rudimentarnu obradu teksta, vredi upoznati još po neku naredbu koja će bitno ubrzati rad. Komande EDT-a se unose preko numeričke tastature koja je, čim startujemo editor, predefinisana tako da odgovara našoj slici. Vidimo da su brojnim tasterima dodeljene po dve funkcije — gornja se izvršava kada pritisnemo sam taster, a donja kada pritisnemo **GOLD** i taster. Vredi primetiti da se taster **GOLD** ovde *ne* ponaša kao **Shift**: da bismo, na primer, izvršili komandu **OPEN LINE** (umetanje prazne linije iza tekuće) pritisnućemo i otpustiti taster **GOLD**, a onda pritisnuti **LINE/OPEN LINE**.

Za početak vredi pritisnuti taster **HELP** i tako dobiti sliku sasvim sličnu našoj — ovaj će nam prizor pomoći kada god zaboravimo lokaciju nekog kontrolnog tastera ili poželimo detaljnija obaveštenja o njemu. Vredi reći da će nam ovakva pomoć biti potrebna razmerno retko — nekoliko osnovnih tastera i mehanizama se lako zapamti, a ostatak se previše retko koristi.

Prvi princip koji ćemo upoznati je kretanje kroz tekst. Tasteri **TOP** i **BOTTOM** zaduženi su za pozicioniranje kurzora na početak i kraj datoteke. **PAGE** pomera kurzor na sledeću stranu (kraj strane je označen sa Ctrl L), **SECT** za 16 redova, **LINE** za jedan red, **WORD** za jednu reč a **CHAR** za jedan znak. Sva pomeranja (osim apsolutnog **TOP** i **BOTTOM**) odvijaju se u pravcu kretanja koji definišemo pritiskom na **ADVANCE** ili **BACKUP**: **ADVANCE SECT** pomera kurzor za 16 redova unapred a **BACKUP SECT SECT** za 32 reda unazad.

Brisanje nepotrebnih znakova, reči i programskih redova izvodimo pritiscima na tastere **DEL C**, **DEL W** i **DEL L** respektivno. Segmenti koje obrišemo sele se u interni bafer, što znači da ih pritiskom na **UND C**, **UND W** odnosno **UND L** možemo vratiti u osnov

ni tekst. Ova se pogodnost odnosi samo na pretodno obrisani znak, reč ili red — novo brisanje nepovratno uništava sadržaj ovog bafera, što znači da ćemo, ako zaključimo da je nepotrebno brisanje izazvalo katastrofalne gubitke, morati da napuštamo editor sa QUIT i tako ignorišemo sve izmene teksta.

Pomeranje i kopiranje teksta obavljamo na principu „makaza i lepka". Tekst koji se pomera najpre treba obeležiti: pozicioniramo kurzor na njegov početak, pritisnemo SELECT i „odvezemo" kurzor do kraja teksta; obeleženi tekst je obično vizuelno istaknut. Zatim pritiskamo CUT i obeleženi tekst u trenu nestaje sa ekrana. Ništa, naravno, nije izgubljeno — tekst je u specijalnom CUT baferu i možemo ga vratiti prostim pritiskom na PASTE! Ovim vraćanjem tekst u baferu neće biti izbrisan niti promenjen, što znači da možemo da pozicioniramo kurzor na neki drugi segment teksta i pritiskom na PASTE, dupliciramo „odsečeni" segment.

Pretraživanje je rešeno na sasvim klasičan način: pritiskamo FIND, kucamo string koji se traži a zatim i ENTER sa numeričke tastature ili FNDNXT. Pošto string bude pronađen, možemo ga na neki način ispraviti i onda pritisnuti FNDNXT u nadi da će string biti pronađen i na nekom drugom mestu. Obzirom da svako pretraživanje počinje od tekuće pozicije kurzora, za analizu čitave datoteke treba pritisnuti TOP i tek onda FIND.

Pomenimo još par korisnih tastera. COMMAND omogućava direktno zadavanje komande linijskog editora bez napuštanja ekranskog (COMMAND EXIT ENTER je, na primer, ekvivalentno sa Ctrl Z EXIT RETURN), <BACKSPACE> pozicionira kurzor na početku tekućeg reda, <LINEFEED> briše reč a Ctrl U segment reda *levo* od kurzora. RESET poništava greškom aktivirano obeležavanje segmenta teksta, SPECINS omogućava umetanje kontrolnih kodova u tekst, REPLACE zamenu stringa i tako dalje. Posebnu pažnju zaslužuje kombinacija tastera Ctrl W koja „osvežava" ekran: u toku rada može da se dogodi da sistem ispiše neku poruku (neki vam, na primer, kolega „telefonira" uz pomoć nalužnog programa PHONE ili je posao koji ste usmerili na paketnu obradu okončan) koja će naoko pokvariti tekst tj. učiniti da tekst na ekranu ne odgovara tekstu u baferu. Ovu nejednakost ispravljamo sa Ctrl W; ako uočite da se precesto nkazuje potreba za ovim tasterom, razmotrite mogućnost da pre startovanja editora otkucate SET TERMINAL/NOBROAD i tako izbegnete razne informativne poruke sistema.

## Datoteka EDTINI.EDT

Pre mesec dana naučili smo da se po uspešno okončanoj autorizaciji korisnika izvršava komandna procedura *login.com* u okviru koje se mogu definisati određene skraćenice i postaviti podrazumevane vrednosti raznih parametara procesa. Sasvim slično tome, po svakom startovanju editora izvršava se editorski program upisan u datoteku *edtini.edt*; u okviru ovog programa možemo da definišemo neke komande i makro naredbe na način koji je detaljno opisan u uputstvu za upotrebu EDT-a. Ovde ćemo samo reći da u *edtini.edt* vredi upisati red koji glasi *set mode change* — na taj će se način editor uvek „buditi" u ekranskom modu.

Nevolja sa datotekom *edtini.edt* je što je editor uvek traži u tekućem direktorijumu — ako radite sa većim brojem kataloga, moraćete da prekopirate datoteku u svaki od njih. Ako vam se ovo rešenje čini nepraktičnim, dopunite *login.com* proceduru redom koji glasi $ edit: = = edit/command=disk:[katalog]edtini.edt

Ovim redom definisana je skraćenica EDIT koja zamenjuje naredbu VMS-a za pozivanje editora i nalaže operativnom sistemu da po startovanju editora potraži datoteku *edtini.edt* na zadatom *disku* i u zadatom *katalogu*.

Opisom programerskog editora završili smo bazični deo ove male škole VMS-a — verujemo da je svaki pažljivi čitalac naših napisa u stanju da komunicira sa bilo kojim VAX-om. Za mesec dana upoznaćemo naredbe rezervisane za upravnika računskog centra i druge „privilegovane korisnike" da bismo u finalnom napisu koji treba da sumira naše ovogodišnje bavljenje operativnim sistemima za veće računare uporedili juniks i VMS i to kako sa korisničke tako i sa programerske strane.

*Dejan Ristanović*





Inteligentno pretraživanje (4)

# Ni krstić ni kružić

**Pošto smo u „Računarima 41" upoznali opšte karakteristike minimaks postupka, pokušaćemo da ga primenimo na igru *krstić-kružić* — videćemo da je stablo ove naoko veoma jednostavne igre prilično široko, što znači da svako „potkresivanje" koje uvedemo dovodi do dramatičnih ubrzanja koja ćemo, jasno, pripisati inteligenciji inteligentnog pretraživanja!**

Igra *krstić-kružić* je oduvek privlačila mlade programere svojom jednostavnošću — obzirom da svaki igrač posle svega nekoliko partija uspeva da nauči kako da dođe do remija, moglo bi se očekivati da će i računaru to jednostavno poći za rukom. Tako nekako i biva — programi za *krstić-kružić* su nekada bili toliko kratki da je i stari TI-57 (pedeset programskih koraka, odnosno stotinak bajtova) zahvaljujući beskrajnom trudu jednog belgijskog programera mogao da bude nepobedivi suparnik! U bivšem „Galaksijinom" Katalogu programa za džepne računare našlo se nekoliko programa uz pomoć kojih su nešto moćniji programabilni kalkulatori (TI-59, HP-41C, sarp PC-1500...) igrali krstić-kružić obezbeđujući i razne luksuzne opcije kao što je vraćanje poteza. A onda se stvar protegla u treću dimenziju — počeli smo da zamišljamo tri table 3×3 koje su se nalazile jedna iznad druge i igrismo „Most" u kome je pobednik igrač koji upiše tri znaka na bilo koji red, računajući tu i razne prostorne dijagonale zamišljene kocke. Za trodimenzionalni *krstić-kružić* smo i emocionalno vezani, pošto je ova igra predstavljena u prvom napisu koji je „Galaksija" posvetila džepnim računarima (januar 1981.), dok je „Prošireni most za TI-59" prvi program koji smo tokom leta 1981. distribuirali u vidu foto-kopiranog listinga.

Na principu DATA lista funkcionišu i ostali trodimenzionalni „Most" programi koje smo tokom 1981. godine opisivali u „Galaksiji".

U okviru ove serije tekstova bavimo se, međutim, nekom vrstom veštačke inteligencije: želimo da program igra „razmišljajući", to jest da poteze pronalazi korišćenjem principa inteligentnog pretraživanja. Takav program za igru *krstić-kružić* je, na žalost, predugačak da bismo ga objavili u „Računarima" (sećate li se ekranskog editora?), pa ćemo ostatak ovog teksta posvetiti nekim opštim savetima i primeru izvršavanja programa — verujemo da će njegovo pisanje predstavljati zanimljiv način da upotrebite slobodno vreme.

## Lažna jednostavnost

Završavajući sa istorijom, autor ovoga teksta mora da prizna da je pre gotovo mesec dana počeo da piše program *krstić-kružić* zasnovan na inteligentnom pretraživanju minimaks postupku opisanom u prethodnim „Računarima" i „potkresivanju stabla" koje ćemo opisati u ovom tekstu. Finalni program na *Turbo Pascalu* bio je dug nekih dvadeset kilobajta, odnosno gotovo duplo duži od prošlomesečnog programa „Osam", u svakom slučaju toliko dugačak da ga nikakva optimizacija ne bi učinila prikladnim za objavljivanje u „Računarima"!

Ostalo je samo da se zapitamo u čemu je greška — jeste da su nas računari PC i računari sa megabajtom RAM-a i hard diskovima od 40 M razmazili, ali je ipak nezamislivo da je za *krstić-kružić* nekad bilo potrebno sto bajtova a danas dvadeset kilobajta. Nije nam ostalo ništa drugo nego da pronađemo TI-59 program za ovu igru i upustimo se u analizu. Ako se izuzme oduševljenje što smo i dalje u stanju da pratimo TI-59 kod, program nam je doneo nekoliko razočarenja: on, pre svega, uopšte ne igra *krstić-kružić!* U program su, da budemo precizniji, ugrađene tabele (danas bismo ih nazvali programske datoteke) u kojima piše kako treba igrati u kojoj poziciji. Obzirom da se u jednoj *krstić-kružić* partiji sreće ogroman broj pozicija (pogledajte prošle „Računare") programeri su primenjivali sjajan trik: kod nekoliko početnih poteza (kasnije je broj mogućih poteza minimalan) table se privremeno rotira tako da popunjen deo dođe u gornji levi ugao. Na ovaj način se brojne simetrične pozicije (pogledajte sliku 3 iz prošlih „Računara" i nemojte se zbuniti što spod te slike piše slika 4 — BIGZ je prošlog meseca uvodio novi foto-slog a radio na starom!) tretiraju zajednički pa je memorija neophodna za „pamćenje" kompletne *krstić-kružić* partija svedena na otprilike 256 bajtova.

## Potkresivanje stabla

O potkresivanju stabla neke igre detaljno smo pisali pre dve godine — da budemo sasvim precizni, na 54. strani „Računara 13". Ovde ćemo ponoviti samo najvažnije činjenice neophodne za pisanje racionalnog *krstić-kružić* programa.

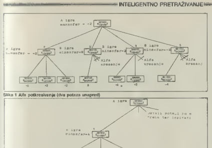
Neporecivа je činjenica da iole složene logičke igre imaju neobično velika i razgranata stabla. U šahu se, na primer, iz svake pozicije u proseku može povući po 30 poteza, što znači da će svaka sledeća generacija poteza koju ispitujemo biti 30 puta veća od prethodne. Obično se kaže da je *faktor grananja* za šah 30.

Ako, dakle, program za šah želi da „gleda" dva poteza unapred, moraće da ispita 1+30+900=931 poziciju. Ako dupliramo broj poteza (četiri unapred) broj pozicija će porasti na 1 + 30 + 900 + 27000 + 81000 = 837931: dvostruko veća „dubina razmatranja" je, dakle, produžila izvršavanje programa celih 900 puta, što čak i najbrži računari današnjice teško mogu da podnesu! Zato je neophodno u startu odbaciti neprosperitetne poteze i ispitivati u veću dubinu samo one koji daju šansu na povoljnu poziciju — to je princip heurističkog pretraživanja. Broj pozicija za ispitivanje, međutim, često biva preveliki čak i ako se primene heuristički algoritmi u koje se može pouzdati — neku granu ponekad treba ispitati do većih dubina, ne zanemarujući ni jedan mogući potez. Metod koji obezbeđuje drastično „kresanje" broja ispitanih pozicija i to bez ikakve opasnosti od zanemarivanja bilo čega značajnog je, verovali ili ne, izmišljen pre gotovo četvrt veka — poznat je pod imenom *alfa beta potkresivanje* ili u stranoj literaturi *alpha beta pruning*.

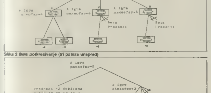
## Alfa . . .

**Ni jedan** program koji pretenduje da pobeđuje u nekoj logičkoj igri **ne može** sebi da dopusti odricanje od tehnike alfa-beta potkresivanja stabla: reč je o metodu koji dovodi do spektakularne uštede u vremenu, a ipak, čak i teorijski, eliminiše rizik da će biti potkresana grana sa najboljim potezom („najbolji" je, jasno, onaj potez koji bi bio otkriven da je datim algoritmom ispitano čitavo stablo do određene dubine).

Upoznajmo najpre alfa potkresivanje posmatrajući jedan sasvim jednostavan primer njegove primene sa slike 1. Vidimo stablo jedne pozicije u igri „Kalah" (previla su opisana u „Računarima 11") ispitiva-



**Slika 1 Alfa potkresivanje (dva poteza unapred)**



**Slika 2 Beta potkresivanje (tri poteza unapred)**



**Slika 3 Alfa-beta potkresivanje (četiri poteza unapred)**

no do dubine 2. Pozicije ispitivane u „prvoj generaciji" su obeležene slovima P, Q, R i S jedino radi jasnijeg i kraćeg izražavanja.

Program prvo detaljno ispituje poziciju P i ustanovljava da je za tu poziciju *minsofar* jednak 2. To znači da je za početnu poziciju *maxsofar* takođe +2.

Ispitujemo sada poziciju Q. Prvi ispitani potez dovodi do pozicije čija je vrednosna funkcija 0. *Minsofar*, prema tome, ne može biti veći od nule ma kakave vrednosti dobili ispitivanjem ostalih pozicija koje proizlaze iz Q. Konačni *minsofar* pozicije Q će, dakle, u svakom slučaju biti manji ili jednak od nule koja je, sa svoje strane, već manja od dosadašnje vrednosti *maxsofar*-a (+2). To znači da mirne duše možemo da odustanemo od daljeg ispitivanja pozicije Q i pređemo na poziciju R. Ovo je *alfa kresanje*.

Alfa kresanje (za trenutak ćemo videti odakle dolazi ovo „alfa") može da se primeni i u pozicijama R i S. Čim vrednost *minsofar* postane manja ili jednaka od vrednosti *maxsofar* u višem čvoru, odustaje se od ispitivanja daljih poteza i prelazi na sledeći čvor.

## Beta potkresivanje

Da bismo upoznali beta potkresivanje, moramo da posmatramo igru bar tri poteza unapred, kao što je urađeno na našoj slici 2; malo smo uprostili crtež prikazujući samo jednu od pozicija do koje se dolazi iz početnog stanja.

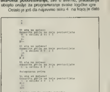
U toku formiranja vrednosti pozicije T program najpre ispituje poziciju U. Ustanovivši da je *maxsofar* pozicije U jednak +4, program postavlja *minsofar* za poziciju T na tu vrednost (4). Zatim se razmatra pozicija V i pri prvom potezu dobija statička vrednost pozicije +6. Pozicija V, dakle, za sada ima *maxsofar* jednak 6. Ma do kakvih rezultata doveli ostali potezi iz pozicije V, njen *maxsofar* će biti veći ili jednak od 6 dok je *minsofar* pozicije T već manji od ovoga broja. Nema, dakle, nikakvog smisla ispitivati dalje poteze iz pozicije V — to je *beta kresanje*.

## Alfa-beta kresanje

Potpuno alfa beta kresanje stabla igre je nešto komplikovanije i daleko moćnije od njegovih komponenti koje smo do sada primenjivali. Upoznaćemo ga posmatrajući sliku 3 na kojoj smo prikazali deo stabla igre koji je analiziran do dubine 4. Pretpostavimo da su leve grane stabla već ispitane i da je tekuće vrednost *maxsofar* za poziciju A jednaka +2. Ispitujući desnu granu stabla, program dolazi do pozicije X koja daje *minsofar* = +1. Mi još nemamo vrednost za *maxsofar* u čvoru neposredno iznad X, ali znamo da početna pozicija stabla ima *maxsofar* jednak +2. Pošto *minsofar* za čvor X ima šanse samo da se smanji ispitivanjem ostalih varijanti, možemo odmah da primenimo alfa kresanje jer ova vrednost nikako ne može da utiče na glavni *maxsofar*. Vrednost pozicije in poteza iznad X može, dakle, da se iskoristi za alfa vrednost koja omogućava „kresanje" u poziciji X.

Uopšte gledano, alfa vrednost za neki čvor je definisana kao maksimum svih *maxsofar* vrednosti čvorova koji su u stablu igre iznad datoga. Slično tome, beta vrednost nekog čvora je minimum svih *minsofar* vrednosti viših čvorova. Vredi još znati da je efikasnost alfa-beta potkresivanja veoma zavisna od redosleda ispitivanja poteza — u „Računarima 13" smo objasnili zašto se ubedljivo najbolji rezultati postižu ako se najpre ispituje najbolji potez (u drugom graničnom slučaju, ispitivanje poteza počevši od najslabijeg bi potpuno ukinulo inače veoma dobro zamišljeno „kresanje"). Sasvim je jasno da na početku ispitivanja ne možemo da znamo koji je potez najbolji (kada bismo to znali, ne bismo ni imali potrebe za ispitivanjem ostalih, zar ne?) ali možemo da se obratimo heurističkim algoritmima — neka nam oni, koristeći neku inventivno zamišljenu vrednosnu funkciju, isporuče potez koji će u najvećem broju slučajeva biti najbolji i potkresivanje će raditi kao sat! U malobrojnim slučajevima u kojima heuristički algoritam da loš rezultat, alfa-beta potkresivanje će, uz izvestan utrošak vremena, ispraviti grešku. Baterija **heurističko pretraživanje / minimaks postupak / alfa-beta potkresivanje**, sve u svemu, predstavlja ubojito oružje za programiranje svake logičke igre.

Ostalo je još da najavimo sliku 4 na kojoj je dato nekoliko primera izvršavanja našeg *krstić-kružić* programa napisanog u skladu sa izloženim idejama; verujemo da ćete ga i sami lako sastaviti. Vreme „razmišljanja" našeg programa izvršenog na AT kompatibilnom računaru nije prelazilo sekund po potezu, kurioziteta radi vredi reći da je ovo vreme bilo šest puta duže pre nego što su primenjeni mehanizmi potkresivanja.



**Slika 4 Primer izvršavanja programa krstić-kružić**

Ovim završavamo našu malu seriju napisa o tehnikama heurističkog pretraživanja: u prvom nastavku smo opisali osnovne ideje, u drugom ih primenili na igru „osam", a u preostale dve proučili kombinovanje heurističkog pretraživanja sa drugim tehnikama programiranja logičkih igara. Zatvarajući drugi tom naše „Radionice logičkih igara", možemo samo da kažemo da se radi o veoma složenoj ali zanimljivoj problematici o kojoj bi se moglo još mnogo pisati: sačekajte „Računare 70"!

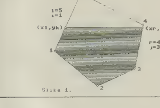
*Dejan Ristanović*

764

Crtanje poligonalne površi

# Na grafičkom poligonu

**Bez poligonalnih površi ne može se ni zamisliti rad programa za crtanje i rad sa trodimenzionalnim telima, jer se složeniji objekti najčešće raščlanjavaju na poligone. Rutina za crtanje poligona može se primeniti na puno mesta — od najjednostavnijih programa za crtanje kontura tela sa skrivanjem nevidljivih linija do programa sa „specijalnim efektima" poput poluprovidnosti.**




Slika 1.

```
CRTANJE POLIGONALNE POVRŠI
Cerovski Viktor (c) 1988

proc poligon
   import y() n
   min=1
   for i=2 to n
      do if y(i)<y(min) then min=i
   l=min  r=l  cntl=0  cntr=0
   repeat
      if cntl=0 then
         j=l  l=l-1  if l=0 then l=n
         setp(j l sxl cl xl,dxl dyl,cntl yk)
      endif
      if cntr=0 then
         i=r  r=r+1  if r>n then r=1
         setp(i r sxr cr xr dxr dyr cntr yk
      endif
      repeat
         line(xl yk xr yk)
         nxtpos(sxl cl xl dxl dy) cntl)
         nxtpos(sxr cr xr dxr dyr cntr)
         yk=yk+1
      until cntl=0 or cntr=0
   until l=r
endproc

proc setp(a b ref(sx c x dx dy cnt yl)
   import x() y()
   dx=abs(x(b)-x(a))  dy=abs(y(b) y(a))
   sx=sgn(x(b) x(a))  x=x(a)  y=y(a)-1
   cnt=max(dx dy)+1
   c=cnt div 2
   nxtpos(sx c dx dy cnt)
endproc

proc nxtpos(sx ref c ref x dx dy ref cnt)
   repeat
      if dy<dx then
         c=c dy  f=FALSE
         if c<0 then c=c+dx  f=TRUE
         x=x+sx
      else
         c=c-dx  f=TRUE
         if c<0 then c=c+dx  x=x+sx
      endif
      cnt=cnt 1
   until f or cnt=0
endproc
```

Poligon (koji određuje površ) zadaje se preko niza tačaka koje predstavljaju susedna temena, a čije su koordinate zapisane u nizovima **x** i **y**. Broj temena poligona je zapisan u promenljivoj **n**. Samu površ ćemo proizvesti iscrtavanjem jedne po jedne horizontalne linije odozdo nagore. Zato je prvo potrebno odrediti redni broj najnižeg temena (promenljiva **min**). Zatim počinje iscrtavanje: promenljive **xl** i **xr** određuju početak i kraj horizontalne duži koja se trenutno iscrtava (y-koordinatu određuje promenljiva **yk**). Početna i krajnja tačka horizontalne duži uvek se nalaze na neke dve ivice poligona. Zato su uvedena još četiri promenljive: par **l**, **j** predstavlja brojeve temena koja su krajevi ivice poligona sa početnom tačkom, dok par **r**, **i** ima sličnu funkciju vezanu za krajnju tačku duži (kao što to pokazuje slika 1).

To, naravno, nije dovoljno da bi se odredile vrednosti **xl** i **xr** — potrebno je izvršiti izračunavanje koordinata (algoritmom za crtanje duži u RA) svake tačke odgovarajućih ivica poligona. Zato su uvedene promenljive **dx, dy, ex, cnt, c** i to dve grupe: jedna se odnosi na početnu, druga na krajnju tačku. Inicijalizacija ovih promenljivih se obavlja procedurom **setp** na sličan način kao kod algoritma za crtanje duži. Promenljive **cnt** stalno imaju vrednost broja neizračunatih tačaka duži — ako je **cnt** dostigla vrednost 0, to znači da je početna tačka tačno na krajnjem temenu ivice pa treba promeniti vrednosti promenljivih **l** i **j** (preći na novu ivicu) a takođe i inicijalizovati **dxl, dyl, exl, cl** i **cntl** na nove vrednosti. U protivnom je dovoljno iscrtati horizontalnu duž, a zatim i izračunati nove vrednosti promenljivih **xl** i **xr**.

Ovo izračunavanje obavlja procedura **nxtpos**. Ova procedura vrši računanje pozicije prve tačke duži koja se nalazi u novom redu. Pri tom ona menja pored odgovarajuće **x** promenljive i vrednosti **cnt** i **c** promenljivih tako da je moguće dalje računati sledeće pozicije pri kasnijim iteracijama. Iscrtavanje vršimo sve dok početna ili krajnja tačka horizontalne duži ne dođe do kraja ivice. Ako pri tom još i promenljive **l** i **r** pokazuju na isto teme to je znak da je površ iscrtana.

Kodiranje ovako dobijenog algoritma u asembleru ne bi trebalo da predstavlja problem, imajući u vidu da algoritam nema ni jednu složeniju operaciju.

Dobijena rutina treba da bude osnovna rutina za iscrtavanje u sklopu nekog programa za rad sa trodimenzionalnim telima. Potprogramu se jednostavno dostavljaju koordinate tačaka poligona (i, naravno, njegova boja). Pri tom treba paziti na redosled iscrtavanja poligona: prvo treba iscrtavati udaljenije poligone pa tek onda one bliže jer se tek tako rešava problem delimične vidljivosti pojedinih poligona (tj. kad projekcija jednog poligona delimično zaklanja projekciju drugog). Pošto program omogućava jednostavnu kontrolu svake tačke površi, lako je moguće modifikovati rutinu tako da popuni poligon proizvoljnim rasterom ili da, na primer, vrši senčenje. Ako se izostavi iscrtavanje nekih tačaka površi može se dobiti efekat poluprovidnosti. Kod jednostavnih vektorski orijentisanih programa ova rutina korisno može da se upotrebi za izbegavanje komplikovanog računa kod delimično vidljivih ivica: kada je dovoljno ovom rutinom obrisati deo ekrana koji zauzima projekcija nekog poligona (naravno pri tom voditi računa o redosledu iscrtavanja), a zatim iscrtati njegovu konturu.

U nekim konkretnim primenama često je potreban rad sa većim brojem malih poligona ili iscrtavanje isključivo trouglova ili četvorouglova. Tada je moguće izvršiti neka pojednostavljenja koja bi za krajnji cilj imala povećanu brzinu rada rutine.

*Viktor Cerovski*

Asembleri (2)

# Tokeni u ringu

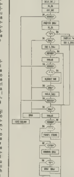
**U „Računarima" je već pisano o stvaranju sistemskog softvera. U nekoliko članaka ćemo pokušati da na konkretnom primeru opišemo nastanak složenog programa — asemblera za mikroprocesor Z80. Na kraju serije ćemo štampati hex-dump i uputstvo za program koji je bio povod za stvaranje ove serije. To je asembler za „amstrad", ali ćemo se truditi da ovi tekstovi koriste svima koji su zainteresovani za programe takve složenosti, a posebno onima koji žele da napišu asembler za bilo koji konkretni mikroprocesor.**

Tokenizacija sažima tekst i znatno pojednostavljuje i ubrzava kasnije asembliranje. Tada neće biti potrebno prepoznavati pojedine mnemonike, nazive registara i prevoditi brojne konstante iz niza cifara u 16-bitnu vrednost, i to u oba prolaza. Treća prednost je što se tokenizacijom automatski ostvaruje i poravnavanje teksta, pošto pri obavljanju inverzne radnje treba samo umetnuti blanke koje će razdvojiti polja za labele, mnemonike, operande i komentare.

## Tokenizacija teksta

Izvršićemo radikalnu transformaciju unetog teksta. Sve službene reči (mnemonici, pseudonaredbe, direktive i nazivi registara) se kodiraju se po jednim bajtom većim od 127. Brojne konstante ćemo predstavljati sa tri bajta: prvi bajt ukazuje da sledeća dva treba tretirati kao jedan broj i određuje da li taj broj treba predstavljati u sistemu sa osnovom 2, 10 ili 16. Zato nije potrebno u tokenizovani tekst unositi i prefikse „#" i „%" koji označavaju heksadekadne, odnosno binarne brojeve. Naravno, linijske brojeve ne treba predstavljati tim bajtom za prepoznavanje jer su oni uvek prva dva bajta u liniji. I labele ćemo predstavljati na isti način, kao što je to već opisano u prošlom nastavku. Druga dva bajta su kod pomoću koga se labele nalaze u tabeli labela. Njihovo tokenizovanje je iskomplikovano time što se imena labela koja još nisu navedena u tekstu moraju ubaciti u tabelu, a kod tokenizovanja ostalih treba uvećati broj u tabeli koji pokazuje koliko se puta takva labela pojavljuju u tekstu. On služi da saznamo kada je neka labela izbrisana iz teksta pa je možemo ukloniti i iz tablice. Ako tokenizujemo labelu koja se prvi put pojavljuje, moramo da za nju nađemo kod koga nema u tablici. U našem programu su za označavanje labela dekadnih, heksadekadnih i binarnih brojeva respektivno upotrebljeni bajtovi #FC, #FD, #FE i #FF.

Sve blanke ćemo izbaciti iz sažetog teksta jer je pri detokenizaciji jednoznačno iz ostalog sadržaja određeno gde ih i koliko treba ubaciti da bi dobijen detokenizovan tekst bio tabuliran. Takođe ćemo eliminisati i karaktere čije je pojavljivanje već određeno onim šta im prethodi — na primer dvotačku posle imena labela kada je ova navedena pre mnemonika. Zareze, zagrade, simbole aritmetičkih operacija, azbučne konstante i komentare ne možemo više sažimati, pa ćemo ih onda jednostavno prepisati. Algoritam po kome naš asembler vrši tokenizaciju je prikazan na slici 1. D_TXT_B i D_TXT_T su adrese bafera koji sadrži tekst unet sa tastature, odnosno u koji se smešta tokenizovan tekst. Pri čitanju se u akumulator prenosi sadržaj (HL), a pri formiranju tokenizovanog teksta se upisuje na adresu (DE). Oba registra se zatim uvećavaju. Ovakvi detalji zbog preglednosti nisu navedeni u algoritmu. Tablice D_TOK1, D_TOK2 i D_TOK3 sadrže spiskove mnemonika, registara i registarskih parova i pseudooperacija asemblera koji se koriste u rutini za tokenizaciju pojedinih reči (pogledaj rutinu TOKEN u rubrici „KLUB Z80" u „Računarima" 38). Ispitivanje KRAJ? utvrđuje kada se došlo do kraja linije (chr$(0)) ili do znaka „;", kada se sve što je preostalo od linije tretira kao komentar i jednostavno prepisuje. Problem tokenizacije sintaksno neispravnih linija ćemo rešiti tako što nećemo dozvoliti da se pojavi. Takve linije će se editovati sve dok ne budu ispravljene ili poništene. Ovakvo presecanje Gordijevog čvora, sem što „humanizuje" asembler i ubrzava programiranje, ima još jednu prednost: pri asembliranju nije potrebno vršiti i proveru sintakse. Međutim, u našoj realizaciji zbog uštede memorije isti potprogram služi za kontrolu sintakse i za prevođenje jedne linije, pa takva lepa mogućnost nije iskorišćena.



Algoritam za tokenizaciju bilo kog seta instrukcija se vrlo malo razlikuje od onog sa slike 1, jer su i formati mašinskih instrukcija za sve mikroprocesore slični.

Algoritam za inverznu operaciju ne bi pokazao ništa suštinski novo, pa ga ovde nećemo prikazati. Osnovni postupak je isti, samo se sada pojedini bajtovi transformišu u reči pomoću istih tablica koje su služile za tokenizaciju. Zgodno je što sada sve službene reči bivaju ispisane malim slovima (možete odabrati i velika, ako mislite da će tekst biti čitljiviji), pa se jasno izdvajaju od ostatka teksta.

Ipak, evo u grubim crtama kako se obavlja detokenizacija. Sadržaj prve dva bajta se ispisuje kao linijski broj, zatim se ispisuje labela ili se, ako tekst ne počinje njome, razmakom popune sva mesta do kolona za instrukcije, ispisuje se mnemonik, opet se ide do sledeće TAB pozicije, ispisuju se operandi i na kraju se od poslednje poravnate kolone prepisuje komentar. Postupak je nešto jednostavniji nego kod tokenizacije jer nema mučenja sa labelama i zato što je sada na ulazu već sređen i sintaksno ispravan niz kodova, dok smo u prvom slučaju morali da predvidimo svaku moguću glupost ili grešku u kucanju.

Bilo bi svakako lakše da smo poštovali zlatno pravilo „Što možeš ostaviti za sutra, zaboravi već danas!" i lepo ostavili labele na miru, kao što svi ostali asembleri i rade. Ali mi zato dolazimo na svoje u kasnijim fazama, kada smo pošteđeni komplikovanog kreiranja tabele simbola i mukotrpnog traženja labela po njoj. O povećanju brzine prevođenja i štednji memorije da i ne govorimo.

## Editor

Izborom EDI poziva operativnog sistema rešili smo problem unošenja teksta sa tastature i njegovog ispravljanja u okviru jedne linije. Preostaje samo da organizujemo ubacivanje, premeštanje i brisanje pojedinih linija ili blokova iz izvornog programa.

Smeštanje jedne linije će se obaviti u nekoliko faza. Uneti tekst se tokenizuje, zatim se vrši provera sintakse (u toku tokenizacije je moguća samo grube provere, jer potprogram koji vrši sažimanje nije u stanju da ustanovi kada instrukcija ima operande koji ne odgovaraju mnemoniku). Vrši se provera da li ima dovoljno memorije i tek sada, ako je sve proteklo u redu, obavlja se tokenizacija sa eventualnim promenama u tablici labela. Zatim se nalazi mesto u izvornom tekstu na koje linija treba da dođe, obezbeđuje se prostor i vrši smeštanje linije. Sav taj posao obavlja rutina ENT_LINE pozvana u listingu glavne petlje iz prošlog nastavka.

Editovanje je pak strahovito jednostavno. Treba samo pronaći liniju, detokenizovati je i taj dobijeni tekst preneti u bafer koji koristi EDI.

Rutinu za rezervisanje mesta u memoriji ćemo napisati tako da se može koristiti za bilo koju lokaciju u radnom prostoru. Problem je u tome što sem premeštanja bloka memorije iznad lokacije zadate za rezervisanje treba korigovati i vrednosti sistemskih promenljivih koje ukazuju na adrese iznad zadate. Na primer, ako se vrši umetanje u izvorni tekst, onda treba uvećati promenljivu koja ukazuje na početak tablice labela (ona se u memoriji nalazi iza izvornog teksta) i promenljivu D_FRSP. Rutine za izbacivanje nekog dela programa ili labele iz tablice mora da obavlja isti posao kod korigovanja sistemskih promenljivih. I uslov za menjanje neke promenljive je isti: potrebno je da ona ima veću vrednost od adrese za umetanje ili brisanje, pa možemo da deo za korekciju izdvojimo u jedan potprogram. Ove rutine su prikazane u listingu 1. M_INS i M_DEL realizuju obe zadatka, a S_WARS vrši zajednički deo posla. Treba primetiti jednostavan trik koji se koristi za umanjivanje promenljivih — registar BC se pre pozivanja SET_WARS komplementira tako da se u kasnijim sabiranjima praktično realizuje oduzimanje. Ulazni podaci u rutine za umetanje i brisanje su adresa gde se rezerviše prostor ili briše (u HL) i broj bajtova (BC).

## Šetanje po tekstu

Većina editorskih komandi se pri izvršavanju odnosi samo na pojedine linije u programu, pa je neophodna rutina za nalaženje adrese linije sa traženim brojem. Na još nižem nivou je potprogram koji pronalazi adresu sledeće linije. To je rutina NEXT_L (listing 2). U slučaju da smo došli do kraja teksta, setuje se flag prenosa. Za pretraživanje od početka treba prvi

**Oslobađanje memorije u radnom prostoru.**
**U HL je adresa, a u BC broj bajtova.**

```
M_INS   push  hl            Sačuvaj HL
        ld    hl, (D_MTOP)  Izračunaj
        ld    de, (D_FRSP)  ;koliko je
        or    a             memorije
        sbc   hl de         ;slobodno
        sbc   hl,bc         ;Uporedi sa traženim prostorom
        ld    hl,TXT_MEMF   Ako nema dovoljno
        jp    c ERR         greška.
        pop   hl            ;Obnovi HL
        res   3, (Iy+1)     Spreči RUN možda se briše
        ld    de, (D_FRSP)  ;objektni program
        ex    de hl
        sbc   hl, de        ,U DE broj bajtova koji se
        ex    de hl         premeštaju
        push  bc
        push  de
        call  S_WARS        ,Podesi promenljive
        ld    hl, (D_FRSP)  ,Premeštanje sa
        dec   hl            D_FRSP—1
        ld    d h
        ld    e l           ,na
        sbc   hl, bc        D_FRSP—1—BC
        pop   bc            Broj bajtova
        lddr                Premeštanje.
        inc   hl            ;Obnovi
        pop   bc            registre.
        ret
```

**Uklanjanje grupe bajtova iz memorije U HL adresa, u BC broj bajtova**

```
M_DEL   xor   a
        sub   c
        ld    c,a

        sbc   a,a
        sub   b
        ld    b,a           ,BC=–BC
        call  S_WARS        Podesi promenljive
        ld    d,h
        ld    e,l           ,Adresa odredišta
        sbc   hl bc         Adresa izvora.
        push  hl
        ld    hl,(D_FRSP)
        inc   hl
        or    a
        sbc   hl,de         Broj bajtova.
        ld    b,h
        ld    c,l
        pop   hl
        ldir                ,Izvrši prebacivanje
        ret
S_WARS  ld    ix,D_SOU      ,Podešavanje sistemskih prom
LOOP1   inc   ix
        inc   ix
        ld    a,(ix)        Upoređivanje
        ld    d (ix+1)      ;vrednosti promenljive
        ld    a,l           ,sa adresom
        sub   e             premeštanja.
        ld    a,h           Ako ukazuje
        sbc   a,d           ,na napomeren prostor,
        jr    nc,TEST       ;sve je u redu
        ex    de hl
        sbc   hl bc         ,Inače je
        ld    (ix),l        ;podesi
        ld    (ix+1) h      ,na novu
        ex    de hl         ;vrednost.
TEST    ld    de D_FRSP
        defb  #DD           ,LD A,XL
        ld    a,l
        sub   e
        defb  #DD           ,LD A XL
        ld    a,h           ,Da li se došlo
        sbc   a,d           ,do D_FRSP?
        jr    c,LOOP1       ,Ako nije, idi opet u petlju
        ret
```

**Listing 1.**

**Rutina koja nalazi sledeću liniju u izvornom programu**

```
NEXT_L  ld    hl (CURRENT)  Tekuća linija je u CURRENT
NXT_L1  inc   hl            Preskoči
        inc   hl            ,linijski broj
LOOP    ld    a (hl)        Uzmi bajt iz linije
        inc   hl
        cp    252           ,Ako je veći od 251
        jr    nc NXT_L1     ;preskoči i sledeća dva
        or    a             Ako se nije došlo do kraja
        jr    nz,LOOP       ;linije traži dalje
TEST_L  ld    c,(hl)        Ispitivanje
        inc   hl            da li postoji sledeća linija.
        ld    b (hl)        Ako su umesto linijskog broja
        dec   hl            dva bajta sa 0
        ld    a,b           ne postoji
        or    c
        ret   z
        ld    (CURR) hl     Inače ona postaje tekuća i
        scf                 setuje se CY
        ret
FRST_L  ld    hl (D_WORK)   Prva linija u radnom segmentu
        jr    TEST_L        ;teksta
```

**Nalaženje linije sa zadatim brojem, ili prve sa većim brojem**

```
FIND_L  push  hl            sačuvaj HL
        ld    d b           ,Broj je bio u BC
        ld    e c
        call  FRST_L        Počni od prve linije
LOOP1   jr    nc EXIT       Ako se došlo do kraja završi
        ex    de hl         Uporedi
        scf                 ;traženi broj
        ld    a,l           ,i
        sbc   a c           ,broj
        ld    a h           ;tekuće
        sbc   a,b           ;linije
        ex    de hl         ,Ako je linija nađena
        jr    c EXIT        ,završi
        call  NEXT_L        Inače idi
        jr    LOOP1         ,na sledeću
EXIT    ex    de,hl         ;Adresa u DE
        pop   hl            ;Obnovi HL
        ret                 I povratak.
```

**Listing 2**

put pozvali FRST_L. U ovom slučaju se na izlazu adresa linije nalazi u HL paru, a njen broj u BC. Često u sklopu linije postoji jedan bajt koji pokazuje koliko je linija duga, pa se nalaženje sledeće linije ostvaruje prosto sabiranjem sadržaja tog bajta i adrese tekuće linije. U našem slučaju nećemo tako postupiti jer bi uvećanje svake linije za po jedan bajt uticalo na utrošak memorije, pa će nam za utvrđivanje početka nove linije poslužiti bajt 0 kojim se predstavlja kraj važeće linije. Pri tome treba preskočiti sve nule koje se pojavljuju u sklopu tokenizovana tabele ili kodirana brojna konstante. Ovaj način traženja je dosta sporiji od prvog ali daje sasvim dobre rezultate, jer ni linije tokenizovanog asemblerskog teksta nisu dugačke — svega nekoliko bajtova. Kraj izvornog teksta ćemo registrovati tako što ćemo posle poslednje linije držati dva prazna bajta, pa će dva bajta posle linije biti ili broj sledeće (koji mora biti različit od nule) ili marker kraja teksta.

Potprogram FIND_L vraća u DE adresu prve linije koja ima isti ili veći broj od onog koji se dostavlja u BC. I u ovom slučaju flag prenosa indicira dolazak do kraja izvornog programa. Obično se komande odnose na liniju sa zadatim brojem ili prvu sa većim, pa FIND_L odgovara za većinu slučajeva. Pošto se na izlazu u BC nalazi broj pronađene linije, mi lako možemo da ispitamo kada tražena linija ne postoji.

Sada je dosta jednostavno realizovati komande editora. Potrebno je samo napisati potprograme za prikupljanje raznih tipova argumenata i onda se pojedine komande ostvaruju korišćenjem već gotovih delova. Naredba LIST se realizuje tako što se pomoću FIND_L utvrdi koji blok linija treba izlistati i onda se linije iz tog intervala detokenizuju i odštampaju na ekranu. Za naredbu za kopiranje dela teksta treba ustanoviti koji se deo kopira i opet pomoću FIND_L naći mesto na koje se prenosi blok, a zatim se tu oslobodi potreban prostor i iskopira sadržaj kome se zatim promena linijski brojevi tako da linije u tekstu i dalje budu u rastućem redosledu. U naredbi za premeštanje ćemo još samo obrisati blok linija koje su prenesene. Slično važi i za ostale editorske komande.

Malo se treba pomučiti samo kod naredbe za pretraživanje i zamenu. Najbolje je ostvariti je tako da se, kada se pronađe linija koja sadrži zadati string, cela linija prikaže sa tim stringom ispisanim inverzno i da se onda čeka pritisak na neku tipku kao odgovor na pitanje „Replace Edit or Next?" Ako je zadat samo prvi string, onda zamena nije moguća, pa se nude samo dve poslednje mogućnosti. Pretraga se vrši sekvencijalnim detokenizovanjem linija iz zadatog intervala i traženjem navedenog dela teksta u baferu koji sadrži detokenizovanu liniju. U slučaju da je string nađen i da treba vršiti zamenu, najjednostavnije je izbrisati ga iz bafera, osloboditi prostor za umetanje druge niske i iskopirati je na to mesto. Tako dobijena linija se dalje tretira kao da je uneta sa tastature — tokenizuje se i smešta u izvorni program na sasvim uobičajen način. Jedino što se posle njenog unošenja ne ide u glavnu editorsku petlju, već se nastavlja potraga.

## Ulaz i izlaz

Pri pisanju asemblera išli smo redom koji nam omogućuje da najlakše proveravamo rad već napisanih delova. Zato smo prvo stvorili editor jer se sve ostale funkcije vrše upravo nad izvornim programom, pa je najprirodnije omogućiti da se stvori tekst na kome ćemo proveravati sve ostale celine. Sada su na redu naredbe za ulaz i izlaz, jer se njima lako možemo da vršimo razne proveravanja na istom tekstu bez njegovog ponovnog kucanja svaki put kada naš asembler krahira. Ove naredbe su najviše zavisne od računara na kome pišemo, pa u ovoj seriji nećemo mnogo ulaziti u detalje. Ali o tome u sledećem broju.

*Branko Marović*

Skupovi na računaru (7)

# Balansirana stabla

**Balansirana stabla izmišljena su samo zato da bi se pri radu sa stablima eliminisali nezgodni slučajevi u kojima se veliki broj elemenata akupa naiazi u jednom podstablu, dok su ostala uglavnom prazna. Stalno uravnotežavanje stabla traži dodatno vreme prilikom dodavanja i uzimanja, ali se zato višestruko isplati pri pretraživanju.**

Svako ko je iole duže koristio stabla za čuvanje elemenata nekog skupa zna koliko je neugodna situacija u kojoj stablo degeneriše u listu pa njegovo pretraživanje postaje katastrofalno sporo. To su posebno iskusili svi oni koji su pokušali da pišu asembler ili bilo koji drugi kompajler.

Neki su se „snalazili" tako što nisu ubacivali elemente u stablo onim redom kojim su nailazili nego su ih čuvali u nekim baferima i kasnije ih ubacivali po slučajnom redosledu. Sva to međutim suviše usporava rad. Ono što je zaista potrebno je postupak koji bi sprečavao stablo da „sklizne" u stranu, to jest održavao njegovu ravnotežu.

## Kako postići ravnotežu

Zamislite da koren stabla predstavlja polugu terazija, a da su levo i desno stablo tasovi. Poluga će biti u ravnoteži ako se na tasovima nalaze iste težine, odnosno isti broj čvorova. Stablo se naziva **perfektno uravnoteženim** (perfectly balanced) ako se za svaki njegov čvor broj čvorova u levom i desnom podstablu razlikuje najviše za jedan. Ovo se može reći i rekurzivno.

Stablo je **perfektno uravnoteženo** ako su mu podstabla perfektno uravnotežena i broj elemenata u njima se razlikuje najviše za jedan.

Najbolje je ipak to videti na primeru. Program na listingu 1 generiše perfektno uravnoteženo stablo od n elemenata.

**listing 1**

Glavna karakteristika perfektno uravnoteženog stabla sa **n** elemenata je to da njegova visina (najduži put korena do nekog lista) nije veća od **NlogN**, pa je to i najduže vreme potrebno da se pronađe neki čvor. Na slici 1 vidite nekoliko perfektno uravnoteženih stabala.



**Slika 1**

Svakim novim dodavanjem ili uzimanjem elementa narušava se ravnoteža pa je potrebno stalno je uspostavljati što je izuzetno „skupo". Niklaus Virt (Wirth) je izračunao da algoritam koji stalno održava stablo u perfektnoj ravnoteži donosi poboljšanje efikasnosti od svega 40 procenata, što je malo za nadoknađivanje dodatnog vremena koje se utroši na uravnotežavanje. Potreban je dakle neki „slabiji" kriterijum uravnoteženosti koji će se lakše zadovoljavati i davati stablo čija visina ne prelazi **NlogN** (za stablo sa **N** elemenata).

Stablo je **uravnoteženo** ili **balansirano** ako se visine podstabala bilo kog čvora razlikuju najviše za jedan.

```
TYPE Link = POINTER TO Node
     Node = RECORD
              num : INTEGER
              left,right : Link
            END;
VAR n : INTEGER; root : Link

procedure BuildTree(n:INTEGER):Link
VAR new : Link
    x,nl,nr : INTEGER

begin
  if n = 0 then return(NIL)
  else
    nl <- n DIV 2; nr <- n-nl-1;
    Allocate(new TSIZE(Node));
    with new^ do
      num <- n; left <- BuildTree(nl) ;
      right <- BuildTree(nr)
    end
  end;
  return(new)
end BuildTree.

procedure PrintTree(t:Link;h:INTEGER);
VAR i:INTEGER
begin
  if t # NIL then
    with t^ do
      PrintTree(left,h+1);
      for i <- 1 to h do
        WriteString("  ")
      end;
      WriteInt(num); WriteLn;
      PrintTree(right,h+1)
    end
  end
end PrintTree.

begin
  ReadInt(n);
  root <- BuildTree(n);
  PrintTree(root,0)
end
```

*listing 1*

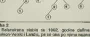
```
TYPE String = ARRAY [0..15] OF CHAR
     Atom = RECORD
              CASE type: BOOLEAN OF
                TRUE: number : INTEGER
                FALSE: name : String
              END;
            END;
     Tlink = POINTER TO Tnode;
     Tnode = RECORD
               bal : INTEGER
               data : Atom;
               left,right : Tlink
             END;
```

*listing 2*

Sada se brojevi elemenata u podstablima mogu razlikovati za mnogo više od 1, a svaki se element i dalje može pronaći u vremenu **O(NlogN)**. Na slici 2 vidite 3 takozvana Fibonačijeva stabla. To je „najgore" balansirano stablo, stablo maksimalne visine (za zadati broj čvorova) koje je još uvek u ravnoteži. Ovo se stablo definiše rekurzivno:

1 — prazno stablo je Fibonačijevo visine 0
2 — jedan čvor je Fibonačijevo stablo visine 1
3 — ako su **T(h—1)** i **T(h—2)** Fibonačijeva stabla visina **h—1** i **h—2** onda je:

**T(h)= T<(h—1), x,T(h—2)>** takođe Fibonačijevo stablo



**Slika 2**

Balansirana stabla su 1962. godine definisali Adelson-Velski i Landis, pa se ona po njima nazivaju i **AVL-stabla**.

Tip podataka za balansirano stablo dat je na listingu 2. Jedina razlika u odnosu na binarno stablo je dodatno polje **bal**, koje sadrži razliku visine desnog i levog stabla (**bal=h(R) — h(L)**) i može imati vrednosti **1,0 i —1.**

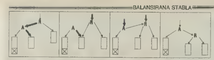
**listing 2**

## Dodavanje elementa

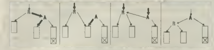
Dodavanje novog elementa AVL-stablu odvija se u dve etape. Prvo se, kao i kod običnog binarnog stabla, pronalazi mesto za novi element. Pretraživanje se izvodi rekurzivno uz „grananje" ulevo i udesno zavisno o tome da li je novi element manji ili veći od „tekućeg" korena. Kad se stigne do lista, novi element se jednostavno „prikači" kao da se radi o najobičnijem binarnom stablu a potom se prelazi u drugu fazu: balansiranje.

Dodavanjem novog elementa mogu nastati tri različita stanja. Radi jednostavnosti pretpostavimo da je novi element dodat levom podstablu (L) nekog korena, povećavši mu visinu za 1. Ako je za dati koren važilo **bal=1,** to jest **h(R) > h(L)** pre dodavanja, posle dodavanja će biti **bal=0,** što će reći da se uravnoteženost poboljšala i ne treba vršiti nikakvo balansiranje. Ukoliko je pre dodavanja bilo **bal=0,** to jest **h(R) = h(L),** posle dodavanja će biti **bal=1,** odnosno perfektna ravnoteža je narušena ali je **AVL-kriterijum** uravnoteženosti još uvek zadovoljen pa ni ovde nema potrebe za bilo kakvim balansiranjem. Jedini slučaj u kome je balansiranje potrebno nastupa ako je pre dodavanja bilo **bal=—1,** to jest h>n < **h(L)**. Posle balansiranja će biti bal=0.

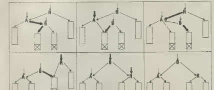
U slučaju dodavanja elementa u levo podstablo koje je već „teže" ( **h(L) > h(R)** ) od desnog balansiranje se izvodi jednostavnom promenom nekoliko linkova. Gledajući stablo na papiru čini se kao da dolazi do rotiranja jednog ili dva para strelica, pa se i niz promene linkova naziva u ovom slučaju **rotacije.** Da bi se odredilo da li će se vršiti jedna ili dve rotacije treba ispitati da li je za narušavanje ravnoteže „krivo" levo ili desno podstablo levog postabla **L.** „Silazi" se dakle jedan nivo dublje i posmatra koren levog podstabla **L.** Ukoliko je njegov **bal=—1,** „krivac" je levo podstablo i treba izvršiti jednu rotaciju koja se shodno „krivcu" (levo podstablo levog podstabla) naziva **(LL)—single** rotacija. U suprotnom slučaju je **bal=1,** pa je „krivac" desno podstablo (levog podstabla L) i treba izvršiti dva rotacije koje se zajedno nazivaju **(LR)—double rotacija.**



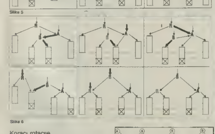
Slike 3



Slike 4



Slike 5

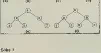


Slike 6

## Koraci rotacije

Na slici 3 vidite sve korake **(LL)—single** rotacije. Prvo se levi pointer korena **R** skrene tako da pokazuje na desno podstablo čvora **A** (na desno podstablo levog podstabla). Potom se desni pointer čvora **A** skrene tako da pokazuje na koren **R**, i konačno se **A** učini novim korenom stabla čime je i balansiranje završeno.

Na slici 4 vidite **(RR)—single** rotaciju koja se izvodi posle dodavanja elementa u desno podstablo koje je već bilo „teže" (to jest, **h(R)** > **h(L)** ), i to u slučaju da je „krivac" za narušavanje ravnoteže bilo desno podstablo desnog podstabla. Rotacija je potpuno simetrična u odnosu na **(LL)—single**, pa je nema potrebe posebno objašnjavati.



Slika 7

Na slici 5 je dat postupak **(LR)—double** rotacije. Prva dva koraka su u stvari koraci **(RR)—single** rotacije, to jest levo podstablo čvora **B** postaje desno podstablo čvora **A**, dok sam čvor **A** postaje levi sin čvora **B**. Sledeća dva koraka predstavljaju, pak **(LL)—single** rotaciju između čvora **B** i korena **R**, odnosno desno podstablo čvora **B** postaje levo podstablo korena **R**, dok sam koren **R** postaje desni sin čvora **B**. U poslednjem koraku čvor **B** postaje i „zvanično" novi koren stabla.

Slika 6 daje **(RL)—double** rotaciju koja se izvodi posle dodavanja elementa u desno podstablo koje je već bilo „teže" ( **h(R)** > **h(L)** ), i to u slučaju da je „krivac" za narušavanje ravnoteže bilo levo podstablo desnog podstabla. Rotacija je potpuno simetrična u odnosu na **(LR)—single**, pa nema potrebe posebno je objašnjavati.

Na slici 7 vidite primer stabla kome se dodaju elementi. Dodavanje broja 7 stablu **(a)** povlači za sobom **(RR)—single** rotaciju, dok dodavanje broja 2 stablu **(b)** ne zahteva nikakvo balansiranje. Dodavanje broja 1 stablu **(c)** zahteva, međutim, **(LL)—single** rotaciju. Dalje dodavanje broja 3 stablu **(d)** zahteva **(LR)—double** rotaciju, a dodavanje broja 5 stablu **(e)** zahteva **(RL)—double** rotaciju da bi se stablo dovelo u red.

## Uzimanje elementa

Znajući kakve je sve komplikacije stvaralo uzimanje elementa iz običnog binarnog stabla, normalno je očekivati ogromne komplikacije pri uzimanju elemenata iz **AVL-stabla**. Na svu sreću to nije tako.

Jedina komplikacija se tiče „otkačivanja" elementa, koje je isto kao i kod običnog binarnog stabla. Ispitivanje narušavanja ravnoteže i samo balansiranje gotovo su isti kao i kod dodavanja elementa. Narušavanje ravnoteže zbog uzimanja elementa iz jednog podstabla može se shvatiti i kao povećanje broja elemenata „suprotnog" podstabla, te se koriste iste rotacije, a samo se testovi za utvrđivanje „krivca" razlikuju.

Pre svega, ako je element uzet iz levog podstabla vrši se balansiranje koje odgovara dodavanju elementa u desno podstablo i obratno. Ravnoteža se narušava samo ako se element uzima iz levog podstabla koje je „lako" (to jest, desno podstablo je „teško", bal=1, odnosno **h(R)** > **h(L)** ) ili iz desnog podstabla koje je „lako" (levo podstablo je „teško"). Tada se, kao i pri dodavanju elementa, ulazi jedan nivo dublje i traži „krivac" za narušavanje ravnoteže, pa se shodno tome izvršava **(LL)—single**, **(RR)—single**, **(LR)—double** ili **(RL)—double** rotacije.

Mada algoritmi za dodavanje i uzimanje elemenata izgledaju vrlo slični, među njima postoji velika razlika. Dok se pri dodavanju izvršava najviše jedna rotacija, pri uzimanju elementa se može izvršiti po jedna rotacija za svaki čvor (koji se) nalazi na putu od korena do elementa koji se briše. Teorijski dakle uklanjanje elementa iz **AVL-stabla** može trajati znatno duže od dodavanja, iako je složenost ista, O(NlogN) za dodavanje N elemenata ili za uklanjanje N elemenata. Praktična merenja pokazuju, međutim, da se jedna rotacija izvrši pri svakom drugom dodavanju, a tek pri svakom petom uklanjanju elementa, što će reći da **AVL-stablo** mnogo bolje „podnosi" uzimanje elemenata nego njihovo dodavanje.

U sledećem nastavku videćete kako rade kompletni algoritmi za **insert** i **Delete** na **AVL-stablu**. Da bi algoritmi funkcionisali potrebno je rediti i dosta „kućnih poslova", tj. brinuti se o tome da pri svakoj rotaciji budu dobro podešena **bal** polja svih čvorova. Pri dodavanju elementa podešavanje praktično ne postoji kod **(LL)—single** i **(RR)—single** rotacija već se vrši isključivo kod **(LR)—double** ili **(RL)—double** rotacija. Prilikom uzimanja elemenata postoji međutim relativno složeno podešavanje balansnih koeficijenata i kod **(LL)—single** i kod **(RR)—single** rotacija no o tome više u sledećem nastavku.

*Žarko Berberski*

# Upoznajte Magmediu da biste je voleli!

## Amerikanci to već znaju. Uverite se i vi.

U želji da upozna jugoslovenske korisnike računara sa kvalitetom ... proizvoda, „Magmedia" je odlučila da u saradnji sa časopisom „Računari" ponudi diskete formata 5.25 i 3.5 inča zainteresovanim obrazovnim ustanovama (školama, fakultetima) i pojedincima sa posebno povoljnim, reklamnim uslovima nabavke.

**Cena za pojedince:**

| | |
|---|---|
| 5.25" DS/DD 48TPI 360 K | 6 500 |
| 3.5" DS/HD 135 TPI 720 K | 14 900 |

**Cene za obrazovne ustanove:**

| | |
|---|---|
| 5.25" DS/DD 48 TPI 360 K | 5 925 |
| 5.25 DS/DD 96 TPI 1 Mb | 7 200 |
| 5.25" DS/HD 96 TPI 1.2 Mb | 9 375. |
| 3.5" DS/DD 135 TPI 720 Kb | 10 875. |

Dovoljno je da popunite odgovarajuću narudžbenicu i da je pošaljete na adresu „Magmedijinog" beogradskog predstavništva
MAGMEDIA TECHNOLOGIES
PIC Beograd
11000 Beograd Ustanička 125c
Obrazovne ustanove su oslobođene plaćanja poreza na magnetne medije pa vas pozivamo da se javite na telefon (011) 489 2323 i 489 4515 radi dogovora. U cene za pojedince je uključen odgovarajući porez.

Uskoro „Magmedia" i „Računari" pokreću akciju u kojoj će čitaoci „Računara" testirati Magmedijine diskete i, uz nagrade, smišljati reklamne slogane za „Magmedijine" proizvode.

Navedene cene „Magmedia" garantuje do 31. oktobra a kvalitet do kraja života.

### Narudžbenica za pojedince

***računari***

Ovim neopozivo naručujem pouzećem sledeće „Magmedijine" proizvode:
____ komada DS/DD 48 TPI 5.25 inčnih disketa po ceni od 6 500 din.
____ komada DS/DD 135 TPI 3.5 inčnih disketa po ceni od 14 900 din.
Pošiljku ću platiti poštaru prilikom preuzimanja.

(Ime i prezime)

(Poštanski broj i mesto) (Ulica i broj)

(Br. lične karte i SUP) (Svojeručni potpis)

### Narudžbenica za obrazovne ustanove

***računari***

Ovim neopozivo naručujem sledeće „Magmedijine" proizvode:
____ komada DS/DD 48 TPI 5.25 inčnih disketa po ceni do 5 925 din.
____ komada DS/DD 96 TPI 5.25 inčnih disketa po ceni od 7 200 din.
____ komada DS/HD 96 TPI 5.25 inčnih disketa po ceni od 9 375 din.
____ komada DS/DD 135 TPI 3.5 inčnih disketa po ceni od 10 875 din.
Prispelu fakturu platićemo u zakonskom roku.

(Puni naziv obrazovne ustanove)

(Poštanski broj i mesto) (Ulica i broj)

M.P.

(Ovlašćeno lice)

**Magmedia — da spremljeno ostane sačuvano**

# računari

## *su uvek aktuelni*

***Ono što nema u novom, potražite u starim brojevima***

- **novi uređaji, testovi, uporedni prikazi**
- **tehnike programiranja, algoritmi, biblioteka programa**
- **komercijalni softver, softverski podsetnici, testovi**
- **najnovije vesti, domaće i strano tržište**
- **klub programera, HELP, bajtovi lične prirode**
- **računari u školi, računari u maloj privredi**
- **akcije: samogradnja „tima 011", samogradnja robota**

## I to se dešava

U nekoliko poslednjih meseci pozivali smo naše nove čitaoce da naruču ju stare brojeve „Računara" — broj 34 i od 36 naviše (brojevi 1—33 i broj 35 već su bili rasprodati). Odziv je premašio naša očekivanja. Na žalost nismo svima uspeli da pošaljemo sve brojeve koje su tražili. Nepažnjom radnika u magacinu, naime, preostala količina brojeva 34, 36, 37 i 38 upućena je u fabriku papira na preradu.

Sve čitaoce koji su naručili neke od navedenih brojeva molimo da uvaže naše izvinjenje. Druge brojeve koje su naručili (između 39 i 42), razume se, već su primili ili će im oni biti upućeni ovih dana. Ovih brojeva ima dovoljno na zalihama a preduzete su mere da se izbegne bilo kakav nesporazum.

## Novi način naručivanja

Da bismo ubrzali proceduru nabavke starih brojeva „Računara" koje još imamo na raspolaganju (od broja 39 do najnovijeg) i smanjili troškove poštarine koje plaća kupac, uveli smo novi način njihovog naručivanja. Procedura je jednostavna:
1. Izaberite brojeve „Računara" između 39 i poslednjeg koje želite da kupite, saberite njihove cene (navedene u pregledu na ovoj strani) i na dobijenu sumu dodajte 1 500 dinara na ime troškova pakovanja, rukovanja i poštarine.
2. Dobijeni ukupni iznos uplatite na žiro-račun 60802-603 23264 (RO BIGZ, Bulevar vojvode Mišića 17, 11000 Beograd) sa obaveznom naznakom da je posredi kupovina starih brojeva „Računara" (navedite brojeve).
3. Potvrdu o uplati ili njenu foto-kopiju obavezno pošaljite na adresu redakcije „Računara" (kao pod 2). Redakcija se obavezuje da će vam naručeni brojevi biti upućeni najkasnije pet dana nakon prijema potvrde o uplati. Za sva eventualne reklamacije odmah se javite redakciji (tel. 011/653 748).

***računari 39*** *cena 1.500*
Hardver: AT baba
Komercijalni softver: MASM, Majkrosoftovi noviteti, modula-2
Algoritmi: hala, inteligentno pretraživanje
Serije: Napravite malog robota
Umetak: MS DOS 3.30

***računari 40*** *cena 2.000*
Komercijalni softver: WordPerfect 5.0, PC TOOLS DELUXE, MuMATH, CED
Operativni sistemi: VMS
Algoritmi: hala, inteligentno pretraživanje
Softverski podsetnik: Art Studio, GEOS 1.3
Umetak: Xerox Ventura Publisher

***računari 41*** *cena 2.000*
Hardver: AT matične ploče, HP Laser Jet S II
Komercijalni softver: WordPerfect 5.0
Modifikacije „atari ST" emulatori
Operativni sistemi: VMS
Tehnike programiranja: Osnovne računarske radnje, inteligentno pretraživanje, Quick Radix Sort
Umetak: Turbo Pascal 4.0

***računari 42*** *cena 2.000*
Komercijalni softver: Actor, XYWrite, Mace Utilities, StatGraphics
Programski jezici: PostScript
Operativni sistem: VMS i kompajleri
Tehnike programiranja: WordPerfect asembler za „amstrad"
Tematski blok: škole — kursevi — seminari računarskog obrazovanja

**računari u većoj školi** cena 1.500
(specijalno izdanje)
Škola bejzika
Sve periferijske jedinice
Zbirka rešenih zadataka iz programiranja
Sve o PC računaru
Samogradnja računara „tim 011"



## Izdavač:

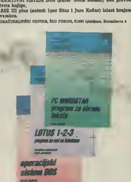
# MIKRORAČUNALNIŠKI CENTER, ŠKD FORUM

**LJUBLJANA, Kersnikova 4, tel: 061/329-185 i 011/142-953**

U okviru biblioteke SODOBNO POSLOVANJE SA PC RAČUNALNIKOM, izašle su iz štampe na srpskohrvatskom jeziku sledeće knjige

- PC WORDSTAR — program za obradu teksta (autor: Nada Rušela)
- LOTUS 1—2—3 — program za rad sa tabelama (autori: Martina Radanovič i Ivan Zaležnik)
- OPERATIVNI SISTEM DOS (autor Boris Butina), dok prevod četvrte knjige,
- dBASE III plus (autori: Igor Sitar i Jure Kočar) izlazi krajem novembra.

MIKRORAČUNALNIŠKI CENTER, ŠKD FORUM, 61000 Ljubljana, Kersnikova 4



**BIBLIOTEKA**
pruža praktična uputstva za rad sa najpoznatijim PC poslovnim programima. Knjige su pisane razumljivim stilom, tako da je mogu koristiti i oni, koji do sada nisu imali mogućnosti da svoj rad osavremene ličnim računarom.

Štampane su na najkvalitetnijoj hartiji, na preko 900 štampanih strana, formata 16×23 cm, moderno grafički opremljene. Knjige su pisane pregledno sa konkretnim primerima i dodacima, a moderan izgled knjiga, nov i savremen pristup određenoj tematici, ovim knjigama daju posebnu vrednost.

**PORUDŽBENICA „Računari 43"**

Ovim neopozivo naručujemo (na srpskohrvatskom jeziku):
1. ________ primeraka knjige PC WORDSTAR po ceni od 22.500 (dvadesetdvehiljadepetstotina dinara)
2. ________ primeraka knjige LOTUS 1—2—3 po ceni od 28.000 (dvadesetosamhiljada dinara)
3. ________ primeraka knjige OPERATIVNI SISTEM DOS, po ceni od 29.000 (dvadesetdevethiljada dinara)

Knjige se plaćaju u gotovom, pouzećem (za privatna lica) ili na žiro račun izdavača (za korisnike društvenih sredstava). U slučaju spora, nadležan je sud u Ljubljani.

________________________________
(ime, odnosno ime prezime)

________________________________
(naziv Radne organizacije)

________________________________
(adresa, telefon i broj lične karte)

________________________________
(mesto i datum) M.P. (potpis ovlašćene osobe)

Narudžbenice obavezno slati na adresu: Agencija „Duga" — SIZS, 11000 Beograd, Bulevar vojvode Mišića 17

---

## TEHNIČKA KNJIGA je najveći jugoslovenski izdavač knjiga o računarima i informatici

Predstavljamo Vam deo ove popularne biblioteke

1. Mr Dragan Pantić
   **LOTUS 1—2—3** (228 str.) — [illegible] 000 d
2. Adem Jakupović
   **dBASE III plus** (207 str.) — [illegible]500 d
3. Armando Jarne
   **TURBO PASCAL** (187 str.) — [illegible] d
4. Časlav Ginić
   **PSCOM 64**
   Hardver PSCOM-a, BASIC kroz primere, Asembler, Računarska radionica (170 str.) — 25 000 d
5. Grupa autora
   **KUĆNI KOMPJUTERI** Algoritmi i programi za Spectrum i Commodore (243 str.) — 28 000 d
6. Grupa autora
   **ŠTA MOŽE COMMODORE 64** (164 str.) — 18 000 d
7. Ian Stewart i Robin Jones
   **COMMODORE 64** — Programiranje na lak način (236 str.) — 13 000 d
8. Grupa autora
   **GRAFIKA I ZVUK ZA COMMODORE 64** (240 str.) — 30 000 d
9. Andrew Bennet
   **MAŠINSKE RUTINE ZA VAŠ COMMODORE 64** (128 str.) — 12 800 d
10. Dejan Ristanović
    **MAŠINSKO PROGRAMIRANJE NA MIKROPROCESORIMA Z80 i 6502** (255 str.) — 18 000 d
11. Mr Veselin Petrović i Zoran Mošorinski
    **COMMODORE 128** (188 str.) — 13 000 d
12. Bob Steele i Jerry Wellington
    **RAČUNARI I KOMUNIKACIJE** (224 str.) — 17 800 d
13. Mr Nenad Marković i Dušan Davidovac
    **ZX SPECTRUM** — Programiranje u BASIC-u (176 str.) — 9 000 d
14. Dr Mirčeta Danilović
    **VIDEO-KOMPJUTERSKE IGRE** (207 str.) — 4 800 d
15. Dejan Stajić
    **INTERFEJSI I MODEMI** (147 str.) — 14 500 d
16. Garry Marshall
    **AMSTRAD CPC 464/664/6128** (120 str.) — 8 200 d
17. Steve Webb
    **AMSTRAD CPC 464** — Programiranje u Asembleru (112 str.) — 8 200 d
18. John Graham
    **LIČNI RAČUNARI** — Vodič za izbor, korišćenje i primenu (270 str.) — 10 800 d
19. Mr Dragan Pantić
    **APLIKACIONI PROGRAMI ZA PERSONALNE RAČUNARE IBM PC AT/XT I APPLE II C** (276 str.) — 14 700 d
20. Mr Vojislav Mišić
    **IBM PC AT/XT U 25 LEKCIJA** (242 str.) — 21 000 d
21. Mr Veselin Petrović i Adem Jakupović
    **LINIJSKI EDITOR ZA SISTEME 6 — HONEYWELL** (207 str.) — 9 800 d
22. Dejan Ristanović
    **OBRADA TEKSTA NA RAČUNARU** (230 str.) — 14 000 d
23. Dr Boško Damjanović
    **BASIC U NASTAVI MATEMATIKE** (114 str.) — 7 800 d
24. Dr Boško Damjanović
    **ZBIRKA ZADATAKA U BASIC-u** (223 str.) — 8 500 d
25. Dr Dušan Tošić i dr Vojislav Stojković
    **PROGRAMSKI JEZIK PASCAL** — Zbirka rešenih zadataka iz programiranja (252 str.) — 16 000 d

## USKORO. AUTO C

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 |

Upišite znak X preko ... koju poručujete. Por... adresu NIRO TEHN... grad, Vojvode Stepe... 031 492-075 ... pouzećem. Knjige se ... nabaviti i u svim vašim ...

Ime i prezime ________

Ulica i broj ________

Broj pošte ________ Mesto ________

**Tehnička knjiga**

Uređuje Dejan Ristanović

# Čikago, brade i orhideje

**Znajući da avgustovski odziv obično nije baš slavan, u „Računarima 41" smo primenili jedan mali trik (takozvani doping): napisali smo da je jedan inače veoma jednostavan zadatak „nešto teži". I — rezultati su tu: sasvim solidnih 195 pisama i svega dva kupona na koje nije bilo upisano ime Luis!**

Podsetimo se, pre svega, zadatka. U malom avionu koji je putovao od Čikaga do Dalasa našle su se tri čoveka: Džejson, Kermit i Luis. Dvojica su članovi Mense, dvojica nose brade, dvojica gaje orhideje, a dvojica su iz Čikaga. Nikoga ne karakterišu više od tri pomenute osobine.

Ako je Džejson član Mense, on je iz Čikaga.

Ako Kermit nosi bradu, on gaji orhideje. Isto se može reći i za Luisa.

Ako je *Džejson* iz Čikaga, on gaji orhideje. Isto se može reći za Luisa.

Pitali smo ko je rođen u Dalasu (možda bi bilo bolje da smo pitali ko nije rođen u Čikagu, ali svima je bilo jasno!) i pružili vam malu pomoć tumačeći rečenicu *Ako je Džejson iz Čikaga, on gaji orhideje* iz koje se *ne* sme zaključiti da Džejson koji (na primer) *nije* iz Čikaga *ne* gaji orhideje!

Počnimo od rešenja koje se dobija bez primene računara, konstruisanog prema prilogu *Petra Kajevskog* iz Skopja: obeležićemo nošenje brade, gajenje orhideja, članstvo u Mensi i stanovanje u Čikagu slovima B, O, M i Č i pokušati da izvedemo neke zaključke. Najjednostavnije bi bilo da je Džejson M a

```
program pitalica [illegible]

Rešenje je [illegible]

Prema programu [illegible]

[illegible]
```

onda bi on bio Č i O dok bi Luis i Kermit bili B pa prema tome i O. Ovim smo došli do protivrečnosti jer se u zadatku tvrdi da orhideje gaje samo dve ličnosti. Dakle, Džejson nije član Mense, pa Kermit i Luis jesu.

Pretpostavimo da Luis i Kermit nose brade i, prema tome, gaje orhideje. Siroti Džejson bi tada morao da bude dve puta iz Čikaga, što je nemoguće, pa znači da Džejson mora da nosi bradu. Uvedimo još pretpostavku da je Luis iz Čikaga i da (prema tome) gaji orhideje — tada bi Džejson bio iz Čikaga, pa bi morao da gaji orhideje, što njemu, kao što smo videli, *ne* pada na pamet. Ostaje da je Luis iz Čikaga.

Što se računarski orijentisanih rešenja tiče, stvar je sasvim jednostavna — kompjuter treba da proveri osamdesetak ili, ukoliko mu baš *ne* malo *ne* pomognemo, par stotina varijanti i izabere onu pravu — sitnice. Zato je prva nagrada pripala programu sa slike 1 koji nas je oduševio jednostavnošću i očiglednošću — delo *Aleksandra Mićica* iz Banje Luke

```
REM
REM        Fibonacijev niz
REM
REM  Prema programu Zorana Rogića
REM
REM        "Računari" [illegible]
REM
[illegible]
početak
  INPUT "Zadati br [illegible]
  [illegible]
petlja
  [illegible]
  IF [illegible] THEN
    [illegible]
    GOTO petlja
  END IF
  [illegible]
  END IF
  IF [illegible] THEN
    PRINT "Generatori [illegible]
    PRINT
    GOTO početak
  END IF
  [illegible]
  GOTO [illegible]
END
```

predstavlja dobar argument u prilog tvrdnji da se pri programiranju *na paskalu* razmišlja u terminima problema koji se rešava, a ne u terminima samog jezika! Što se brzine tiče, program se na našoj zvaničnoj mašini izvršio za 0 sekundi — primili smo i par bržih (!?), ali je vreme jednostavno beznačajno!

Preostale dve nagrade izvukli smo iz grupe tačnih odgovora: 40 000 dinara pripalo je *Željku Kučan* iz Banje Luke (još se nije dogodilo da prve dve nagrada odu u isto mesto, ali ... to je Fortuna) a 20 000 dinara *Draganu Grbiću* iz Sremčice. Čestitke zaslužuju i *Milomir Aleksić, Vladimir Blažić, Zoran Stojanović, Dragan Toroman* i *Davorin Žugčić*.

*Ostalo je još da odgovorimo na pitanje brojnih čitalaca koji su želeli da znaju kako smo smislili imena junaka ove Pitalice — za Džejsona i Luisa još i nekako, ali Kermit ... (zavisno od stepena profesionalne deformacije Kermit će vas podsećati na ime jednog žapca ili na beskrajne minute čekanja da se neki podaci prebace na relaciji VAX — PC). Razlozi su brojni, ali nam prostor u „Računarima" dopušta da navedemo samo jedan — radi se o originalnim imenima iz Mensinog zadatka!*

## (Skoro) svaki broj

Istorija je pokazala da u ovoj rubrici zabrljamo u proseku jednom godišnje. Džoker za 1988. potrošili smo u „Računarima 39" kada smo objavili program koji za zadati element, određuje generatore Fibonačijevog niza. Program zaista radi korektno za traženi broj 1 000 000, ali za nekih petnaestak procenata drugih brojeva ne daje optimalno rešenje. Zahvaljujući se svim čitaocima koji su nam skrenuli pažnju na ovaj propust, na slici 2 objavljujemo korektan program koji je sastavio *Zoran Rogić* iz Ljubljane.

Verujemo da je ova greška dobra prilika da opišemo tretman primedbi na rešenja pojedinih pitalica. Ukoliko se pokaže da je izabrano rešenje ili objavljeni program bio pogrešan ili nedovoljno tačan, objavljujemo ispravku. Prostor u „Računarima" nam, na žalost, *ne* dopušta da komentarišemo naknadno primljena programa koji su nešto kraći ili nešto brži od objavljenih: ovakve priloge, međutim, pažljivo čuvamo u nadi da ćemo ih jednoga dana upotrebiti — možda od pitalica i (upravo obnovljene) Eureke u „Galaksiji" jednom nastane knjiga.

## Preko 100

*Šta kažete za jednu laku pitalicu iz oblasti kombinatorike? Za početak prebrojte sve dvocifrene proste brojeve manje od 45 (11, 13, 17, ... 43), a onda prebrojte različite (zbirovi 11+13+17 i 11+17+13 su isti) načine na koje tri ili više ovih brojeva mogu da daju zbir strogo veći od 100 (dve moguće kombinacije su 37+41+43 i 11+23+29+41+43) i upišite njihov broj na naš kupon ili njegovu fotokopiju koju ćete, zajedno sa programom koji ste koristili da dođete do rešenja, poslati na adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd, tako da pristigne **pre 25. oktobra 1988.** Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 60 000, 40 000 i 20 000 dinara, dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica. Identifikacioni broj dobijete tako što u prve tri kućice upišete **poslednje** tri cifre nečijeg broja telefona, a sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koja šaljete u toku godine nose vaš identifikacioni broj.*

```
36: Od prostih brojeva 11, 13, ... 43 moze se konstruisati
    ____________ razlicitih zbirova vecih od 100.

Ime i prezime    ______________________________
Adresa           ______________________________
Mesto            ______________________________
Kompjuter        ______________  Vreme ________

Identifikacioni broj.  |__| |__| |__|   |__| |__|   |__| |__|
                       iz broja tel.    god.rodj.   po izboru
```

# Bajtovi lične prirode (4)

**Četvrta epizoda naših „Bajtova lične prirode" je u pravom smislu posvećena čitaocima „Računara": „bajtove" čiji su redni brojevi 30, 31, 32 i 40 dugujemo respektivno *Vladanu Dugariću* iz Beograda, *Petru Petroviću* iz Maribora, *Arnoldu Finderleu* iz Rijeke i *Miodragu Vukušiću* iz Titograda. Posebno nam je drago što su se našli odgovori praktično na sva pitanja koja smo postavili pre tri meseca.**

30 **[PJ PASCAL].** Pravi programeri govore fortran, „zdeisči pile" paskal. Jedna od veličanstvenih stvari koje fortran nudi je naredba EQUIVALENCE — uvedete racionalnu promenljivu A i celobrojnu promenljivu B, dodelite im isti memorijski prostor (EQUIVALENCE (A, B)) i onda bez problema pristupate mantisi i eksponentu racionalnog broja i izvodite razne slične čarolije. Isto je stvar verovali ili ne ras položive i na paskalu, i to zahvaljujući promenljivim slogovima. Slika 1 daje savršenu ilustraciju primene tehnike — program učitava ceo broj sa tastature, a zatim ga ispisuje kao racionalan! Iako je program testiran uz pomoć Borlandovog *Turbo Pascal* kompajlera, može se izvršiti i na nejstandardnijem ISO Pascalu.

```
program skora_kao_fortran (input, output);
type flag = Boolean;
     slog = record
              case flag of
                 true:  (a: integer);
                 false: (b: real);
              end;
var sl:   slog;
begin
   repeat
      readln (sl.a);
      writeln (sl.b)
   until false
end.
```

31 **[INT 8, PC, ST].** Sećate li se problema u prenosu podataka na relaciji IBM PC — „atari ST" POMENUTOG U „Računarima 40"? Iako sličan, PC-jev i „atarijev" format nisu baš identični, „atari" upotrebljava duže FAT tablice (sa funkcionalnog stanovišta malo bitno) i ne upisuje neke za PC važne podatke u BOOT sektor i FAT1. To je dijagnoza, a sada sledi rešenje u dva koraka: (1) Bajtove 0, 1 i 2 u BOOT sektoru (staza 0, sektor 1) promenite u &EB, &34 i &90 (zavisno od verzije DOS-a koju koristite, možda će biti potrebne vrednosti &ED, &27 i &00 — pokušajte); (2) Bajt nule sa početka FAT1 tablice (staza 0, sektor 2) promenite sa &F7 na &F9. Otkucajte DIR i „Atarijeva" disketa se pretvorila u PC-jevu. Vredi još reći da se na ovaj način mogu konvertovati samo diskete snimljene u standardnom formatu (2 strane, 80 traka, 9 sektora na traci), „atarijev" TOS prepoznaje i neke druge formate, na primer 83 trake sa 10 sektora po traci.

32. **[TEHP 5, PC]:** Kako će program pisan na nekom popularnom jeziku (na primer *Turbo Pascalu*) saznati u koji je katalog upisan? Pomoću procedure koja je ilustrovana na slici 2. Tajna je u tome što svaki program po učitavanju u memoriju dobija zaglavlje zvano PCP ili *program segment prefix* u koje je pored ostalog upisano i ime datoteke. Pretpostavka je jedino da koristite neku od novijih (3.XX) verzija MS DOS-a.

```
program ko_sam_ja (output);
function ime: string;
var env, i: integer;
    temp:   string;
begin
   env:=memw[prefixseg:$2C]; i:=0;
   while mem[env:i] <> 0 do
         i:=i+1;
   i:=i+4; temp:='';
   while mem[env:i] <> 0 do
   begin
      temp:=temp+chr(mem[env:i]);
      i:=i+1
   end;
   ime:=temp
end;
begin
   writeln ('Zovem se: ', ime);
end.
```

33. **[KOMS, PC]:** Jedan od manje poznatih postaća paketa *Norton Utilities* je program NCD (*Norton Change Directory*). Jedan od kataloga vašeg hard diska se na primer zove */tekst/wperf/printer* a vi ste trenutno u katalogu */razvoj/probe*. Možete da otkucate *cd /tekst/wperf/printer* ali i jednostavno *ncd printer* pa čak i *ncd pri*. Kucanjem same komande NCD na ekranu se pojavljuje grafički prikaz stabla kataloga koji omogućava interaktivno pozicioniranje u bilo koji od njih. Posle instalacije programa NCD kucanje džinovskih imena direktorijuma i mnoštvo obrnutih kosih crta predstavljaće samo neprijatnu uspomenu!

34. **[KOMS, PC].** Mana programa NCD je što posle svakog kompaktiranja hard diska (npr. pomoću programa COMPRESS iz paketa *PC Tools De Luxe*) morate da obrišete datoteku TREEINFO.NCD u osnovnom katalogu (*del/treeinfo.ncd*), a zatim da otkucate jedno NCD. Nije strašno!

35 **[STAM, LJET2]:** *Hewlett-Packardov LaserJet II* teorijski posmatrano ima nekoliko ugrađenih fontova. Ovaj broj se praktično svodi na dve pisma (*Courier i Line Printer*) i nekoliko njihovih varijacija: polucrno, razni setovi znakova, dve orijentacije. Odlučili ste da napišete bejzik programčić koji će se obraćati direktno laseru i birati jedno od pisama. Mislite da se pismo bira nekom kratkom sekvencom? Taman posla — posle dugotrajnog prelurenja po uputstvu i tehničkom priručniku doći ćete do „kilometarskih" sekvenci koje su ilustrovane na našoj slici 3.

```
10 REM
20 REM Fontovi ugradjeni u LaserJet II
30 REM          (PC karakter set)
40 REM
50 REM             "Racunari 43"
60 REM
70 ESC$=CHR$(27)
80 REM Courier 10 pitch normal
90 LPRINT ESC$;"&l0O";ESC$;"(10U";
100 LPRINT ESC$;"(s0p10h12v0s0b3T"
110 LPRINT "Courier 10 pitch normal"
120 REM Courier 10 pitch bold
130 LPRINT ESC$;"&l0O";ESC$;"(10U";
140 LPRINT ESC$;"(s0p10h12v0s3b3T";
150 LPRINT "Courier 10 pitch bold"
160 REM Line Printer normal
170 LPRINT ESC$;"&l0O";ESC$;"(10U";
180 LPRINT ESC$;"(s0p16.6h8.5v0s0b0T";
190 LPRINT "Line printer normal"
200 LPRINT CHR$(12);
210 END

Courier 10 pitch normal
Courier 10 pitch bold
Line printer normal
```

Posle ovih sekvenci čovek se jednostavno zapita ko je ovde lud: korisnik ili *Hewlett Packardovi* programeri. Tajna je u tome što se delim sekvencama precizno opisuje pismo: kakva mu je orijentacija, kolika su slova, koliko su pojačana, koje je pismo... Štampač će po nekom svojim kriterijumima izabrati pismo koje je najsličnije traženom. Ovakvo zadavanje bi bilo dobrodošlo kada bi se u ROM u krilo nekoliko stotina fontova pa ih korisnik ne bi mogao popamtiti: lakše je napisati da su potrebna bold slova visoka 10 tačaka nego pamtiti da se radi o fontu 211! Pošto su na raspolaganju samo dva fonta kilometarska sekvenca su pravo prevcalo opterećenje čak i za srećne vlasnike kertridža.

36 **[STAM? LJET2]** Sekvence iz prethodnog primera mogu se skratiti kombinovanjem: veliko slovo pre svakog ESC treba zameniti malim čime se sledeći ESC izbegava.

37 **[STAM, LJET2]:** Da li vam je na raspolaganju neki neproporcionalni soft font za *LaserJet II?* Rado bismo objavili nekoliko primera.

38 **[OTXT WPERF, PC],** *WordPerfect* 5.0 omogućava editovanje jednom definisanih makroa. Ukoliko na primer želite da editujete makro PROBA pritisnućete *Macro Def* (Ctrl F10), zatim rać PROBA i, najzad, dva puta izabrati opciju 2 iz menija. Prvi problem nastaje kada u makro naredbu treba ugraditi komandu poput HOME ili ENTER — pritisak na odgovarajući taster zapravo predstavlja neku editorsku komandu koja se izvršava a ne ugrađuje u makro. Rešenje je prečica Ctrl V — da biste umetnuli HOME kucate Ctrl V HOME.

39 **[ZAST, PC]:** Ume li neko da razbije zaštitu na igrama *Space Quest 2, Moebius* i *Defender of the Crown? Copylipc 4.01* je jednostavno nemoćan.

40 **[PJ, FORTH, A800]:** *Fig forth* za „atari 800 XL" učitava se u BLC modu i odmah aktivira editor. Iz editora se definišu i isprobavaju nove reči, ali se posle svake „mrtve petlje" čitav postupak mora ponoviti od početka. Zato je zgodno posle dodavanja nekoliko novih reči snimiti novu verziju forta sa CSAVE — snimak je nezavisan od BLC-a i otporan na reset, što znači da će se iz nedestruktivnih petlji lako izlaziti. Mana je jedino u tome što se reči definisane pre kucanja komande CSAVE više ne mogu obrisati sa FORGET.

41 **[STAM, EPSON]** „Računari" su u svojoj dugoj i slavnoj istoriji objavili brojne tabele od kojih su neke nadmašile stalni stanovnici vašeg stola. Urednik ove rubrike ubedljivo najčešće koristi jednu naoko beznačajnu tabelu iz umetka „Štampači" („Računari 16"). Tabela koju repriziramo na slici 5 sumira aktiviranje i deaktiviranje raznih tipova slova na Epson kompatibilnim matričnim štampačima. Pokušaćemo da za sledeće „Bajtove lične prirode" pripremimo pristojnu PC kompatibilnu ASCII tabelu.

IZBOR I OPOZIV FONTOVA
(Epson FX/RX/LX)

| Mod | Izbor | Opoziv |
|---|---|---|
| Condensed | CHR$(15) | CHR$(18) |
| Double strike | ESC "G" | ESC "H" |
| Elite | ESC "M" | ESC "P" |
| Emphasized | ESC "E" | ESC "F" |
| Enlarged | ESC "W" 1 | ESC "W" 0 |
| Italic | ESC "4" | ESC "5" |
| Proportional | ESC "p" 1 | ESC "p" 0 |
| Subscript | ESC "S" 1 | ESC "T" |
| Superscript | ESC "S" 0 | ESC "T" |
| Underlined | ESC "-" 1 | ESC "-" 0 |

42 **[TERM]:** Za potencijalne kompjuterske polliglote: na ruskom se hardver kaže *apparatnoje časti* a softver *matobespečanje*. Memorija je *pamjat*, ali ne pitanje da li se kontinualna memorija zove *večnaja pamjat* ne umemo da odgovorimo.

Priloge za ovu rubriku šaljite na adresu „Računari" (za „Bajtove lične prirode") Bulevar vojvode Mišića 17, Beograd.

*Uređuje Dejan Ristanović*

# Mali oglasi

## Važno obaveštenje

Od ovog broja [illegible] se na bilo koji način nudi bespravno [illegible] softver i [illegible] kopirana literatura. Novac za unapred plaćene oglase za [illegible] u ovom broju biće uredno vraćen pošiljaocima. Detaljnije [illegible] ovog redakcijskog stava i odluke pružićemo u sledećem broju

## SPEKTRUM

REKLAME pomoću spektruma. Program primenjen na jugoslovenskoj turističkoj berzi, sajmovima, hotelima. Telefon 071/ 514-487

## IBM

Diskete DS/DD, 5.25, 4000 i 4500 din. Enisa, tel. 071/214-319

## HARDVER

Prodajem ZX Spectrum sa „trenel" tastaturom, džepni kompjuter Sharp PC 1261 tel. 078/232 836

Commodore PC 128 sa kasetofonom po volji prodajem. Tel 013/311 102.

PC XT turbo sa hard diskom, ocarinjeno, prodajem. tel. 041/217 487

Prodajem Commodore interfejs za štampač Star NL-10 sa uputstvom i dva kabla. Tel. 024/42-826

## RAZNO

KORICE za pojedina godišta vaših „računara" — 6000 din., tel. 011/102 124

Neophodan deo računarske opreme!

Filter za otklanjanje smetnji iz mreže, jačine 800 W. Zbog učešća deviznih komponenti (2 VDR-a, 2 ferita jezgra) štampana ploča sa zalemljenim elementima do 1. XI isporučuje se pouzećem za samo 19 000 — din.

Milanković — Jovčić, Beograd, Proleterskih brigada 12/4, tel. 337-436

**COMPUTER SERVICE**
**41000 ZAGREB**
**VIII VRBIK 33a/VI**

**TEL. (041) 539-277**
**od 10 do 17 sati**

**— brz i kvalitetan popravak vaših računara,**
**— prodaja rezervnih dijelova.**

**ZX SPECTRUM**
- centronic interface
- interface za 1 joystick;
- interface za 2 joystick-a
- input/output interface
- memorijska proširenja,
- tv kabel,
- ear kabel,
- folija za tastaturu.

**ATARI**
- memorijska proširenja,
- TOS u ROM-u,
- disk drive 3 1/2"
- disk drive 5 1/4"
- centronic kabel
- scart RGB kabel,
- video kabel.

**COMMODORE**
- eprom moduli u kućištu s reset tipkom (TURBO 250, T.T. II, PICA, SIMON'S BASIC itd)
- SPEED DOS,
- centronic kabel
- tv kabel,
- video kabel
- audio kabel.

**AMSTRAD**
- interface za 2 joystick-a,
- centronic kabel.

Da biste bili u dodiru sa superinteligentnima, čitajte oktobarski broj „Galaksije"

| | |
|---|---|
| Vazduhoplovstvo | TUPOLJEV NA HIDROGENSKI POGON |
| Kontrola letenja. | JEZA BLISKIH SUSRETA |
| Tema broja. | EKSPLOZIJA ŽIVOG SUNCA |
| | STRAH OD RADIJACIJE |
| | ZAGONETNE SUNČEVE PEGE |
| Futurologija. | PRVO NASELJE NA MESECU |
| Nanotehnologija. | BESKRAJNA SVEST |
| Raketna tehnika. | „NESPOSOBAN" ZA MATEMATIKU |
| Kosmonautika. | BONDARENKO PRVA ŽRTVA |
| Kosmologija. | VELIKOG PRASKA NIJE BILO |

„Galaksija" se družila sa superinteligentnima, saznaćete ko su najpametniji Jugoslaveni

Ostanite u dodiru sa promenama, družite se sa „Galaksijom"

Samogradnja „tima 011“

# Tim dobro, tim bolje

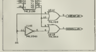
**Akcija samogradnja „tima 011“ ulazi polako u završnu fazu — krajem septambra isporučena su i poslednja štampana kola, a nakon kratkotrajnog šoka zbog visokih cena čipova — nismo mi izmislili inflaciju, devalvaciju i američko-japanski rat memorija — krenula su i narudžbenica za delova iz inostranstva. Očekujemo vaše rezultate.**

Sve one koji su se odlučili da svoj „tim 011“ opreme PC tastaturom, a takvih će biti najviše čeka još malo hardverskih napora. U čemu je problem?

Interfejs za PC tastaturu generiše signal prekide INTO/O a procesor ga opslužuje u modalitetu 2. Način potvrde prijema prekida procesora HD64180 prouzrokuje nepravilan rad registra za pomeranje slike (SCROLL).

```
R8 ..... 150 ohm , 1/4 W
R9 ..... 1 K ohm , 1/4 W
R10 ...  KRATKO-SPOJNIK
R11 .... 1 k ohm , 1/4 W
C6 ..... 33 pF KERAMIČKI
Y1 ..... 12.288 MHZ
........................
R13 .... KRATKO-SPOJNIK
R14 .... 1 K ohm , 1/4 W
R15 .... 150 ohm , 1/4 W
R16 .... 1 K ohm , 1/4 W
C7 ..... 33 pF KERAMIČKI
Y2 ..... 6.000 MHZ
........................
```

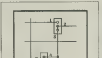
Tabela 1 — Komponente za komercijalnu verziju kvarcnog oscilatora



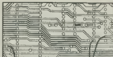
Slika 1 — Detalj šeme pre ...



Slika 2 ... i posle izmena za PC tastaturu



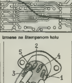

Podešavanje kratkospojnika



Izmene na štampanom kolu



„Tetris“ Ivana Gerenčira




Konektor za tastaturu

Problem se rešava kidanjem nožice 9 i 11 kola U14 (74LS240) pre njegove ugradnje i povezivanjem odgovarajućih lemnih tačaka na matičnoj ploči sa neiskorišćenim nožicama 6 i 4 kola U23 (74LS00). Nožica 5 kola U23 spaja se sa vodom INTO/O. To je sve ... elegantno, zar ne?

Ova izmena nije neophodna za one koji se odluče da koriste postojeću ili neku drugu tastaturu koja računaru predaje ASCII kodove. Jedini problem u ovom slučaju nastaje ako vaša tastatura ne radi na 9600 boda sa 8 bitova za podatke i parnim paritetom. U tom slučaju za početak pošaljite nam tehnička podatke vaše tastature i sistemsku disketu i mi ćemo vam rado pomoći. Kasnije ćete moći da se obratite bilo kome u svom susedstvu ko poseduje „tim 011“ u radnom stanju da vam pomoću programa za konfiguraciju računara (CONFIG) snimi potrebne parametre serijskog porta 1 na sistemsku disketu.

Na slici 1 prikazan je detalj šema računara pre opisane izmene, a na slici 2 posle. Na slici 3 nacrtano je kako se izmene vrši na matičnoj ploči računara.

Većina PC tastature nema ugrađene tzv. PULL UP otpornike na izlazima za takt i podatke. Na žalost oni su postojali na tastaturi koju smo koristili za razvoj interfejsa, te je propuštena njihova ugradnja na pločicu. Zato predlažemo da ova dva otpornika vrednosti 2,2 kohm povežete direktno na petopolni DIN konektor. Da ne dođe do zabune, na slici 4 obeležene su brojevima nožice konektora i odgovarajuće rupice na pločici iterfejsa koje se spajaju žicama.

Još jedan detalj. Pošto smo u poslednjem trenutku odlučili da vam uskratimo rešenje sa jednim integrisanim kvarcnim oscilatorom koje zahteva dodatnu pločicu, u tabeli 1 dajemo vrednosti diskretnih komponenata za oscilatore od 12.288 MHZ i 6 MHZ (16 MHZ za FD9268). Srećno sklapanje!

Lično iskustvo uspešnog probnog sklapača „tim“-a (dobrovoljca) biće objavljeno u sledećem broju „Računara“.

### Kako dalje

Verovatno se kao budući korisnici „tim-a" pitate da li će se za njega pojaviti i ostali programski jezici kao na primer C, MODULA2, ada, fortran, bejzik prevodioci itd. Moramo da se pohvalimo da već sada raspolažemo izuzetno kvalitetnim programskim jezicima koji su preneti na „tim-011" od kojih su C, modula2 i multiprogramski BASIC prevodilac napisani tako da mogu da koriste i ceo Mbajt koji adresira HD64180 u verziji sa 68 nožica. U toku je i razvoj novog grafičkog paketa koji će se koristiti uz ove programske jezike, a biće u stanju da podrži više ekrana prozora i sprajtove. Potrudićemo se da i vi što pre dođete do ovog softvera.

Od hardversko-softverskih projekata uskoro očekujete miša za TIM-011. Ovaj projekat uključuje i proširenje operativnog sistema sa interaktivnim podsistemima za menije i pomoć kao i rutinama za kontrolu SHELL podsistema.

Zatim sledi projekat interfejsa za HARD DISK.

Želja nam je da na TIM-u afirmišemo i neke od mogućnosti računara u industrijskim primenama.

Sa nestrpljenjem očekujemo i vašu podršku i saradnju.

### Sistemska disketa

Kada srećno sklopite vaš „tim-011" i uključite ga, na ekranu ćete dobiti poruku.

Poruka na ekranu su rezultat izvršavanja startne ALIAS komande START. Da objasnimo „Tim-011"

# Komponente iz inostranstva

Svi potrebni delovi za samogradnju „tima" su podeljeni u pet kompleta čije spiskove možete videti u prilogu. Komplete možete naručivati pojedinačno, odjednom ili postepeno, onako kako vama odgovara. Pri tome treba da se pridržavate nekih jednostavnih pravila:

— deviznu uplatu sa svog ili deviznog računa svojih rođaka možete da izvršite na adresu prodavnice i kopiju potvrde o uplati morate da pošaljete uz narudžbu

— važe i pismene narudžbe uz broj kreditne kartice American Express

— nije potrebno tražiti nikakav predračun (profakturu)

— kreditna kartica mora da glasi na ime onoga kome će roba biti upućena (ako umesto vas uplati rođak ili prijatelj, naručeni kompleti biće upućeni njemu)

— obavezno nam pošaljite registracioni listić, koji nam je potreban radi evidencije učesnika u akciji, a vama može da donese povlastice u daljnjoj saradnji.

Ujedno vas obaveštavamo da je do daljnjeg obustavljeno primanje uplata za štampane pločice. Ako ste i dalje zainteresovani, a još niste uplatili, molimo vas da se javite redakciji na telefon 653-746 (pozivni broj za Beograd 011).

Programiranje EPROM-a počelo je 15. septembra. Treba da pošaljete isključivo vaše EPROM-e na adresu redakcije RAČUNARI, Bulevar vojvode Mišića 17, 11000 Beograd. Ispravan i prazan EPROM treba da zabodete u provodni sunđer, stavite u praznu kutiju od šibica i sve to zajedno sa potvrdom o uplati pošaljete vrednosnim pismom na adresu redakcije. U protivnom ne garantujemo za sigurnost vaše pošiljke. Nadoknada za programiranje EPROM-a i poštanski troškovi iznose 5 000 dinara i uplaćuje se na žiro račun BIGZ-a 60802-603-23264 uz naznaku „RA — programiranje EPROM-a".

Nadamo se da će vaš „tim" brzo proraditi na našu i vašu radost.

**Cena i sadržaj kompleta za samogradnju „tima 011"**

**Komplet 1**
Procesor, disk kontroler, EPROM, statički i dinamički RAM, kristali i podnožja za osetljiva integrisana kola.
Cena 2250.80 ATS, pakovanje i poštarina 136.00 ATS, ukupno 2386.80 ATS

**Komplet 2**
Ostala integrisana kola, otporne dekade, blok kondenzatori, konektori i višežilni kablovi za povezivanje
Cena 717.70 ATS, pakovanje i poštarina 136.00 ATS, ukupno 853.70 ATS

**Komplet 3**
Disketni pogon 3.5 1 MB FD 1037 NEC i konektor za napajanje
Cena 1471.00 ATS, pakovanje i poštarina 136.00 ATS, ukupno 1607.00 ATS

**Komplet 4**
Tastatura PC XT standard 84 tastera
Cena 760.00 ATS, pakovanje i poštarina 136.00 ATS, ukupno 896.00 ATS

**Komplet 5**
Monohromni monitor 12"
Cena 1529.00 ATS, pakovanje i poštarina 191.00 ATS, ukupno 1720.00 ATS

Ukupna cena 6738.50 ATS, pakovanje i poštarina 735.00 ATS, ukupno 7473.50 ATS.

**Još jednom svi potrebni podaci: Uplata ili narudžbina kreditnom karticom**

MLAKAR UND CO
Export — Import
Unterbergen 82
9163 Unterbergen
Austria

Ako u banci imate problema sa uplatom na adresu, ona se može izvršiti na konto broj 114-264962 kod banke **Bank Fuer Kaernten un Steiermark, Ruprechter str. 55, 9020 Klagenfurt, Austria**

**Lična kupovina za gotovo ili kreditnom karticom**
Prodavnica se nalazi na glavnom putu Ljubljana, Kranj, Tržič, granični prelaz Ljubelj, Klagenfurt, Minhen, 12 km od graničnog prelaza Ljubelj

**Za pozive iz Jugoslavije**
Telefon 9943-4227-2333
Telefaks 9943-4227-2081
Teleks 422749

**Registracioni listić**
Računari 43 — „tim 011"

Obaveštavam vas da sam dana ........ naručio sledeće komplete:

1 2 3 4 5

Uplatu sam izvršio
1. kreditnom karticom
2. gotovinski na adresu prodavnice
3. karticom odnosno gotovinski u samoj prodavnici

Ime i prezime ..................

Ulica i broj ..................

Broj pošte i grad ..................

Listić poslati na adresu: Računari, Bulevar vojvode Mišića 17, 11000 Beograd

**N A R U D Ž B E N I C A**
Računari 43 — „tim 011"

Molim Vas da mi pošaljete sledeće komplete za samogradnju „tima"

1 2 3 4 5

1. Komplete sam uplatio na vašu adresu, o čemu svedoči kopija bankovne uplatnice koju vam šaljem u prilogu
2. Molim vas da za vrednost poručene pošiljke teretite moju kreditnu karticu American Express broj .................. koja važi do ........ (zaokružite broj ispred načina plaćanja koji koristite)

Ime i prezime ..................

Ulica i broj ..................

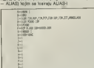
Broj pošte i grad .................. Yugoslavia

Narudžbenicu poslati na adresu: Mlakar & Co. Export-Import, 9163 Unterbergen, Austria



Ekran 1

radi pod CP/M kompatibilnim operativnim sistemom Z. Jedno od njegovih proširenja u odnosu na CP/M je i mogućnost zadavanja više komandi odvojenih tačka-zarezom (;) u istoj komandnoj liniji. Sasvim normalno je da uz ovo postoji i program koji ovako združene komande pamti u datoteku čije se ime proizvoljno zadaje i kreira tzv. ALIAS komande (ALIAS — drugo ime). Analizirajmo malo sadržaj ALIAS-a START pomoću programa VALIAS (Video — ALIAS) kojim se kreiraju ALIAS-i.



Ekran 2

Komanda MDSK I vrši inicijalizaciju i brisanje RAM — diska (označenog slovom M:). Komanda EMU pokreće emulaciju terminala. Ovde se treba podsetiti strukture Z sistema o kojoj smo ranije pisali. Z sistem se sastoji iz više delova. To su: BIOS zadužen za podršku hardvera, ZRDOS koji procesira informacije, ZCPR3 koji procesira komande koje vi zadajete i više tzv. dinamičkih paketa operativnog sistema. Ovi paketi se nazivaju dinamički zato što se mogu u toku rada zameniti drugima. Na taj način moguće je izmeniti osobine operativnog sistema prema potrebama korisnika ili aplikacije. EMU predstavlja jedan od tih paketa koji je priključen operativnom sistemu da softverski omogući funkcije tekstualnog terminala. Pored njega tu su TIM RCP paket rezidentnih (stalno prisutnih) komandi, TIM FCP paket komandi za kontrolu toka izvršavanja svih ostalih komandi, TIM IOP paket ulazno-izlazne podrške periferijskih uređaja računara, TIM Z3T paket sa parametrima terminala i NAMES NDR paket sa imenima tzv. imenovanih direktorijuma (kataloga sa imenima disk datoteka). Svi ovi paketi se priključuju operativnom sistemu pomoću punioca (Loader) LDR. Komanda LDR PCKBD IOP služi da zameni ulazno-izlazni paket TIM IOP novim paketom PCKBD IOP u kome se nalazi podrška PC tastature. Korisnici PC tastature gube mogućnosti koje sadrži paket TIM IOP, a to su fizičko-logička redirekcija periferijskih uređaja i kopiranje sadržaja ekrana na štampač pomoću kojega su dobijene sve slike ekrana u ovom prikazu. „Krupan gubitak!" zavapiće oni koji ne poseduju TIM-011 tastaturu. Za utehu — ovo je samo privremeno rešenje softverske podrške PC tastature i treba da popuni vremenski period potreban za definitivno rešenje (koje zahteva korenite izmene operativnog sistema) u koje će biti uključeno i niz do sada „navođenih" podobljšanja operativnog sistema. Onima koji se snađu i obezbede TIM 011 kompatibilne tastature savetujemo da iz START ALIAS-a odmah uklone linije D i E. Na liniji E nalazi se komanda PCKBD koja inicijalizuje rad PC tastature. Sve dok se one na izvrši „nije zdravo" pritiskati tastere na PC tastaturi pošto to dovodi do sigurnog „pada" računara. Na linijama F i G nalaze se komande koje služe da omoguće rad DOS editora. On služi za ispravljenje pogrešnih i izvršavanje prethodno unetih komandi. Ako vam se ne dopada, možete ga odstraniti brisanjem linije G ALIAS-a START. Na kraju, komanda KBD YSHC služi za zadavanje parametara rada TIM tastature.

Po izvršavanju komande START poruka A:SYSTGEM› označava da operativni sistem očekuje vaše komande. Komande Z operativnog sistema mogu se po mestu izvora podeliti na komande ugrađene u ZCPR3 komandni procesor, komande rezidentnog paketa TIM RCP i TIM FCP i tranzijentne komande koje se nalaze u datotekama na disku. Njihova upotreba znatno je olakšana postojanjem ONLINE-HELP-a (ugrađene pomoći) koje se dobija ako se iza imena komande otkuca blanko i dve kose crte (//). Ovo važi za sva komplikovanija tranzijentna komande. Komanda H lista naziva komandi iz paketa TIM RCP, a pomoću programa SHOW (pokaži) mogu se pored ostalog videti imena komandi paketa kontrola toka TIM FCP. Kompletno uputstvo za korišćenje Z operativnog sistema nalazi se u knjizi koja se uskoro očekuje u izdanju Građevinske knjige. Ovu knjigu vam preporučujemo kao obaveznu TIM 011 literaturu. U sledećim redovnim napisima potrudićemo se da vam potpuno osvetlimo rad u mašinskom jeziku, upotrebu svih sistemskih poziva, a potrudićemo se da dođete i do izvornih verzija operativnog sistema.

U ovome članku ograničićemo se na kratko navođenje komandi, uslužnih programa i datoteka koje se nalaze na vašoj disketi. Uz objašnjenje svake komande/datoteke u zagradama se nalaze oznake sledećeg značenja:

A — ALIAS komanda
I — Interaktivno izvršavanje
P — Ugrađena pomoć
R — Rezidentna komanda
S — Sistemska datoteka
T — Tranzijentna komanda

**Katalog SYSTEM**

ABC — Zadaje ASCII set znakova karakter generatora. (T)
BOOT — Vrši softverski reset sistema. Učitava se operativni sistem sa diska. (T)
CD (Change Dir) — Menja naziv direktorijuma (kataloga programa). (A)
CIR — Zadaje ćirilični set znakova karakter generatora. (T)
CLEANDIR — Čisti direktorijum od nepotrebnih informacija. (TP)
CLS — Briše ekran. (R)
COMMENT — Šalje komentar na ekran ili štampač. (TP)
CONFIG — Menja konfiguracione parametre periferijskih uređaja. (TI)
CP — Kopira datoteke uz mogućnost promene imena. (R)
CRC — Ispisuje CRC (Cyclic Redundancy Check) datoteka na ekranu ili u datoteci. Služi za utvrđivanje da li su datoteke identične. (TP)
DINI — Inicijalizuje disk sistem. (R)
DCP (Disk Copy) — Kopira diskete upotrebom jednog diska. (TP)
DEV — Vrši redirekciju periferijskih uređaja. (TP)
DIR — Prikazuje sadržaj direktorijuma. (R)
DFA (Display File Attributes) — Prikazuje atribute (oznake) datoteka. (TP)
DIFF (Difference) — Prikazuje razlike datoteka. (TP)
DOSED I — Inicijalizuje rad DOS editora. (TI)
DU3 (Disk Utility) — Uslužni program za pretraživanje i menjanje sadržaja podataka na disku. (TPI)
DUMP — Prikazuje sadržaj datoteke ASCII i heksadecimalno. (TP)
ECHO — Izdaje uneti tekst na ekran ili štampač bez intervencije operativnog sistema. Koristan je za slanje kontrolnih i komandnih nizova znakova. (R)
ELSE — Kontroliše tok komandi vezano uz IF. (R)
EMU — Paket za emulaciju terminala. (S)
ERA — Briše datoteke. (R)
ERASE — Briše datoteke uz niz opcija. (TP)
FBAD (Find Bad) — Pronalazi i izoluje loše blokove na disketi. (T)
FI — Završava delovanje prethodnog IF-a. (R)
FINDF (Find File) — Traži datoteke po svim direktorijumima i diskovima. (TP)
FVC — Formatira, verifikuje i kopira diskete. (TI)
GET — Učitava sadržaj datoteke u TPA. (R)
GO — Izvršava program koji se već nalazi u TPA uz prosleđivanje parametara iz komandne linije. (R)
GOTO — Bezuslovni skok u programu ZEX komandnog procesora. (TP)
H — Ispisuje komande sadržane u paketu rezidentnih komandi. (R)
IF — Komanda uslovnog grananja komandnih programa (ALIAS i ZEX). (R)
JUMP — „Skače" na navedenu adresu. (R)
KBD — Zadaje parametre rada TIM-011 tastature (KeyBoarD). (TP)
LAT — Zadaje latinični set znakova. (T)
LDR — Pridružuje datoteke paketa Z sistemu. (TP)
LIST — Ispisuje sadržaj datoteke na štampač. (R)
MCOPY — Kopira datoteke uz niz opcija. (TP)
MDSK I — Inicijalizuje RAM-disk. (TP)
MKDIR (Make Dir) — Zadaje, menja sadržaj paketa imenovanih direktorijuma. (TPI)
MOVE — Premešta (pomera) datoteke iz jednog direktorijuma u drugi. (TP)
MU3 (Memory Utility) — Uslužni program za pregledanje i promenu sadržaja memorije. (TPI)
NAMES.NDR — Paket imena imenovanih direktorijuma. (S)
NOTE — Komanda bez dejstva za pisanje komentara u ALIAS i ZEX programima. (R)
P — Ispisuje sadržaj dela memorije na ekran. (R)
PATH — Zadaje imena direktorijuma koji se automatski pretražuju. (TP)
PCKBD — Komanda za inicijalizaciju paketa PCKBD IOP. (T)
PCKBD IOP — Ulazno-izlazni paket za podršku PC tastature. (S)
POKE — Upisuje novi sadržaj u memoriju. (R)
PRINT — Ispisuje sadržaj datoteke na štampač. (TP)
PROTECT — Postavlja datotekama zaštitne atribute. (TP)
PWD (Print Working Directories) — Ispisuje imena direktorijuma. (TP)
REG — Prikazuje korisničkim registrima koji su deo operativnog sistema. (R)
REN — Menja ime datoteke. (R)
RENAME — Menja ime jedne ili više datoteka uz opcije. (TP)
SAVE — Zapisuje na disketu sadržaj TPA. (R)
SHOW — Ispisuje razne podatke o trenutnom stanju sistema. (TP)
START — Inicijalizuje koja se vrši neposredno po učitavanju operativnog sistema. (A)
SYSGEN — Formira zapis operativnog sistema na disketi. (TP)
TDIR — Prikazuje sadržaj direktorijuma uz dodatne informacije. (T)
TIM.FCP — Paket komandi kontrole toka. (S)
TIM IOP — Paket rutina za redirekciju ulaza-izlaza. (S)
TIM RCP — Paket rezidentnih komandi. (S)
TIM.Z3T — Paket sa parametrima terminala. (S)
TIME — Prikazuje časovnik. (TP)
TPASIZE — Ispisuje veličinu TPA. (T)
TYPE — Ispisuje sadržaj datoteke na ekran. (R)
UNERASE — Restaurira izbrisane datoteke (ukoliko je moguće). (TP)
VALIAS — Uslužni program za kreiranje ALIAS-a. (TPI)
VFILER — Uslužni program za rad sa datotekama. (TPI)
VTYPE — Ispisuje sadržaj datoteke na ekran uz mogućnost pretraživanja. (TPI)
WS (Word Star) — Uslužni program za obradu teksta. (TI)
WSMSGS OVR — Datoteka neophodna za rad WS-a.
WSOVLY1 OVR — Datoteka neophodna za rad WS-a.
XDIR — Prikazuje sadržaj direktorijuma uz još više opcija i informacija. (TP)
XIF — Završava delovanje svih prethodnih IF-ova. (R)
ZEX — Procesor komandnih programa. (TPI)
ZSYSTEM MDL — Zapis operativnog sistema koji se komandom SYSGEN upisuje na sistemske trage diskete. (S)

**Katalog BASIC**

GBASIC COM — Basic interpreter proširen grafičkim instrukcijama.
DEMO.BAS — Demo program na BASIC-u
Sadržaj direktorijuma PASCAL
GRDEMO PAS — Pokazuje upotrebu grafičkih procedura.
GTURBO COM — ALIAS koji instalira grafičku biblioteku i pokreće Turbo Pascal.
PASGR.COM — Paket grafičkih rutina za PASCAL. (Nije kompletan)
PASGR INC — Definicije grafičkih procedura
PRIME.PAS — Demo program.
TURBO.COM — Turbo Pascal
TURBO MSG — Datoteka neophodna za funkcionisanje Turbo Pascala.
TURBO OVER — Datoteka neophodna za funkcionisanje Turbo Pascala.

**Katalog ASEMBLER**

MLOAD — Pretvara datoteku sa INTEL HEX sadržajem u izvršni program.
PORTS LIB — Biblioteka podataka o ulazno-izlaznim portovima ugrađenim periferijama procesora HD64180
SYSLIB REL — Biblioteka sa preko 300 sistemskih potprograma
TEST Z80 — Spisak svih instrukcija procesora Z80 u formi asemblerskog programa
VLIB REL — Biblioteka potprograma za upravljanje ekranom.
Z3LIB REL — Biblioteka potprograma za iskorišćavanje specifičnosti Z operativnog sistema.
ZAS — Relokacioni makro asembler
ZCON — Pretvara izvorni mašinski program pisan za procesore Intel 8080 ili 8085 u Z80 izvorni program
ZDM — Debugger
ZLIB — Program za manipulaciju bibliotekama (library manager)
ZLINK — Linker

Nenad Dunjić

PET PLUS

Uređuje: Nevenka Spalević

Metodika nastave računarstva (2)

# U znanju je moć

**Seriju tekstova o metodici nastave informatike i računarstva započetu u prošlom broju nastavljamo tekstom posvećenim prvim časovima vežbi na računaru, na kojima učenici treba da se upoznaju sa programskim jezikom bejzik. Naš predlog je da se u prvom bloku od 6 časova obrade sledeće teme: Upoznavanje programskog jezika bejzik, elementarne i složene konstrukcije jezika; Upoznavanje strukture bejzik interpretatora, komande unošenja novog programa, ispravke programa, izdavanja, snimanja i učitavanja programa sa kasete ili disketa; Neposredan i programski režim rada; Zadaci za računanje u oba režima rada.**

**OSNOVNI CILJ.** Utvrđivanje prvih saznanja o bejziku stečenih na teorijskim časovima i sticanje prvih iskustava u radu sa bejzik interpretatorom.

**ZAHTEVI ZA ZNANJA I UMEĆA.** Učenici treba da steknu predstavu o izgledu bejzik programa i nauče pravila za zapis različitih tipova konstanti i promenljivih. Stoga je potrebno da već u ovom trenutku umeju da koriste jednostavnu formu naredbe PRINT i naredbe dodele i u direktnom i u programskom režimu rada. U ovom bloku časova učenici takođe treba da nauče da koriste osnovne komande interpretatora, da startuju gotove programe i razumeju značenje najčešćih poruka interpretatora.

**METODSKA UPUTSTVA.** Pre nego što počne praktičan rad s računarom potrebno je napraviti razliku između fizičkog i logičkog reda (linije) i između neposrednog i programskog režima rada.

Fizički red je niska simbola fiksne dužine. Da li fizički red čini 40, 80 ili neki drugi broj simbola zavisi od karakteristika računara i izlaznog uređaja. Na primer „spectrum" izdaje tekstove u fizičkim redovima dužine 32 simbola, dok „tim 011" ispisuje na monitoru 80 simbola u redu, a na štampaču odgovarajućih karakteristika može izdavati informacije i sa po 240 znakova u jednom fizičkom redu.

Logički red predstavlja niske simbola koje se završavaju simbolom „Prelazak u sledeći red", to jest simbolom sa aski kodom 13, koji se pri radu sa bejzik interpretatorom ne vidi na ekranu. Logički red može sadržati više fizičkih redova, ali po pravilu ne više od 255 znakova.

Bejzik interpretator je sistemski program koji omogućava da računar, koji „razume" jedino mašinski jezik, izvršava naše zahteve zapisane u bejziku, višem programskom jeziku koji je relativno lak za učenje. On može da radi u dva režima: neposrednom, u kome odmah interpretira svaki logički red, i programskom, u kome instrukcije zadate u okviru logičkih redova pamti kao programske redove u za to posebno izdvojenoj zoni operativne memorije — zoni bejzik programa. Uz program se čuvaju i podaci koje on koristi. Odgovarajuća oblast memorije zove se zona promenljivih. Informacija po kojoj bejzik interpretator zna da li treba odmah da izvrši naš zahtev ili treba da ga upamti je broj programskog reda. Naime, ukoliko je zahtev započet cifrom, bejzik interpretator čitavu poruku tretira kao programski red i smešta je u zonu programa očekujući nove zahteve. Na primer, ako napišemo

10 PRINT A

na ekranu ništa neće biti odštampano. Ali ako zahtev nije počeo brojem, odmah će biti analiziran i, ako nema sintaksnih grešaka, izvršen.



Da bi se pokazalo kako se računar može u neposrednom režimu rada koristiti kao veoma moćan kalkulator, neophodno je objasniti upotrebu i dejstvo jednostavnih mogućnosti naredbe dodele i naredbe PRINT. Uz to se kroz primere ilustruju i tipovi konstanti i promenljivih u bejziku.

**PRIMER 1.** Zapisati u neposrednom režimu rada (korišćenjem naredbe dodele i naredbe PRINT) sledeće brojeve:

a) 8,0, b) 7 4567 c) 0.0000034 d) 0 000000000005, e) 1 0000060001, f) 4•10 ↑ 15 g) 8 35•10 ↑ −8

(Napomena: Simbol ↑ predstavlja stepenovanje.)

**PRIMER 2.** Korišćenjem računara u neposrednom režimu rada pronaći greške u zapisu sledećih numeričkih konstanti na bejziku:

a) 7 8 b) 63 4−E2, c) 5 0878545E2 d) 8 04E−1 8 e) 8,4E−18 f) − 400000055E3.

**PRIMER 3.** Izračunati u neposrednom režimu rada vrednosti aritmetičkih izraza

a) 7 5•10 ↑ −2+a−4/d,

b) $8+x-\frac{4,1}{2+8}$

c) (8,3+a)•(3,2−8),

d) $\frac{x^2-x+2}{x^4-5x^2+20}$

za sledeće vrednosti promenljivih x=78, a=2 5 b= 56E−3 d=−0.002, f=8. Izraze zapisati u listi PRINT naredbe. (Napomenuti da se naredbe PRINT ne mora pisati slovo po slovo, već je dovoljno otkucati znak pitanje, koji predstavlja njenu skraćenicu u većini dijalekata bejzika.)

Da učenici ne bi stekli pogrešnu predstavu da se računari koriste isključivo za računanje, već na samom početku treba pokazati da su u stanju da obrađuju i tekstove. Bejzik jezik, uz numeričke, može da operiše i sa znakovnim konstantama. Reči „cica", „maca" i bilo koja druga niska simbola između navodnika za računar predstavlja znakovnu konstantu. Između navodnika možemo pisati sve znake — slova, cifre, razmake i sve ostale na ekranu vidljive i nevidljive simbole sa tastature. Kako se i nad znakovnim konstantama mogu vršiti operacije, može se ilustrovati u neposrednom režimu sledećim primerom:

PRINT „cica"+„maca"

što kao rezultat daje zbir navedene dve znakovne konstante.

Za ilustraciju programskog režima rada i korišćenja promenljivih može se koristiti sledeći program koji učenici treba sami da otkucaju:

```
10 A=1.23456789
20 B%=1 23456789
30 C$=1 23456789
40 PRINT A B% C$
```

Treba napomenuti da se za brojeve programskih redova ne koriste uzastopni prirodni brojevi zbog lak-

šeg umetanja eventualnih dopuna u program. Uz ovaj primer treba objasniti mogućnosti komande LIST, načina ispravki grešaka u programu i mogućnosti komande RUN, posle čega se mogu komentarisati činjenice vezane za korišćenje promenljivih u bejziku.

Kada je ovim programom svim promenljivim dodeljena ista vrednost, biće odštampana tri različita broja. Promenljiva A ima vrednost 1.23457, jer se u običnoj preciznosti radi sa 6 značajnih cifara. Promenljiva B% iznosi 1, jer je to ceo deo unetog broja, a promenljiva C# iznosi tačno onoliko koliko je u liniji 30 zapisano 1.23456789, jer se u dvostrukoj preciznosti pamti čak 16 značajnih cifara. (Ovo važi za računare „tim 011")

Kada se govori o promenljivim, uz pravila o imenovanju promenljivih, treba razjasniti još dva pojma: tekuću vrednost i oblast definisanosti. Vrednost promenljive odgovara sadržaju memorijskog prostora koji je dodeljen promenljivoj sa odgovarajućim imenom. U našem primeru promenljiva B% ima vrednost 1. Oblast definisanosti je skup svih vrednosti koje bi mogle biti dodeljena promenljivoj, sve vrednosti koje mogu da „stanu" u odgovarajući memorijski prostor. Tako su celobrojnim promenljivim dodeljena dva bajta i to namenski samo za ceo deo broja. Kada pokušamo da tu smestimo broj 1.23456789 vidimo da može da stane samo njegov ceo deo, a razlomljeni biva „odsečen". Ako pak pokušamo da na isto mesto stavimo broj 33000, to neće uspeti čak ni sa „odsecanjem", jer je broj prevelik. Ova situacija zove se prekoračenje i računar je prijavljuje sa „Overflow". Međutim, pokušaj da se promenljivoj A dodeli broj 33000 ne stvara nikakve probleme, jer broj 33000 pripada oblasti definisanosti promenljivih obične tačnosti.

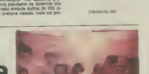
Slično važi i za znakovne (azbučne) promenljive. Ako azbučnoj promenljivoj pokušamo da dodelimo bilo koju nisku simbola dužine do 255 između znakova navoda, neće biti problema. Ali pokušaj dodele A$=1 ili nekog teksta dužeg od 255 znakova neće uspeti.

**LITERATURA**. Za pripremu ovog bloka časova, uz udžbenik korisno mogu poslužiti „Zbirka zadataka iz informatike i računarstva" od M. Stojanovića i V. Aleksića u izdanju „Građevinske knjige" (vidi prikaz) i specijalno izdanje našeg časopisa „Računari u školi". Mi smo uz ove izvore koristili i metodska uputstva objavljena u sovjetskom časopisu „Informatika i obrazovanie" 5/87.

(Nastaviće se)

Test za oktobar

# Od abakusa do računara



Test iz osnova informatike i računarstva koji ovde objavljujemo pripremili smo na osnovu pitanja i zadataka za vežbu iz udžbenika „Računarstvo s programskim jezikom paskal" Ivana Bratka i Vladislava Rajkoviča. Ova knjiga, koja može dosta pomoći u savladavanju osnova informatike i računarstva, može se nabaviti u „Nolitovim" knjižarama po ceni od 3.600 dinara. Odgovore dajemo na kraju testa.

1. Objasnite pojmove računarstvo, informatika i kibernetika.
2. Kakva je razlika između algoritma i programa?
3. Razmislite o procesu dorade nekog predmeta na strugu i recite koja je faza dorade najsličnija *unošenju programa* u računar, a koja odgovara *unosu podataka*:
   a) postavljanje na strug predmeta koji želimo da obradimo,
   b) postavljanje noževa za obradu,
   c) uklanjanje obrađenog predmeta sa struga.
4. Razmislite o merenju napona voltmetrom i recite šta predstavlja „izlaz" tog merenja:
   a) izbor mernog područja,
   b) priključenje napona koji merimo na instrument,
   c) skretanje kazaljke.
5. Koje su od obrade podataka navedenih u sledećim primerima numeričke, a koje nenumeričke:
   a) izračunavanje korena kvadratne jednačine,
   b) sabiranje tri broja,
   c) traženje najdužeg prezimena u registru stanovništva,
   d) uređivanje rečnika,
   e) prevođenje s jednog jezika na drugi,
   f) obračun telefonske pretplate.
6. Koji od sledećih uređaja pruža diskretan, a koji neprekidan zapis informacije:
   a) vodomer,
   b) manometar,
   c) pisaća mašina,
   d) registar-kasa,
   e) merač nivoa goriva u automobilu.
7. Koji su od sledećih zapisa informacije neprekidni, a koji diskretni?
   a) zapis pomoću dijagrama informacije o kretanju temperature bolesnika,
   b) tabelarni zapis vrednosti funkcije sin(x),
   c) zapis vrednosti logaritma u logaritamskim tablicama,
   d) zapis informacije o kretanju vazdušnog pritiska pomoću barografa.
8. Kako povećavamo tačnost pomoću digitalnog, a kako pomoću analognog zapisa nekog broja?
9. Koja od sledećih računarskih naprava upotrebljava analogni, a koja digitalni princip računanja?
   a) stoni kalkulator,
   b) logaritmar,
   c) registar-kasa,
   d) abakus.
10. Koji su od navedenih jezika prirodni, a koji veštački?
    a) paskal,
    b) engleski,
    c) jezik hemijskih formula,
    d) jezik aritmetičkih izraza,
    e) kineski.
11. Koji su od navedenih jezika programski jezici?
    a) srpskohrvatski,
    b) fortran,
    c) engleski,
    d) jezik aritmetičkih izraza,
    e) paskal,
    f) simbolički jezik,
    g) jezik hemijskih formula,
    h) mašinski jezik,
    i) prolog.
12. Zašto program za prevođenje ne može da prevede program sa sintaksnim greškama?
13. Od čega nas oslobađaju neproceduralni jezici?
    a) od svakog programiranja,
    b) od programiranja na mašinskom jeziku,
    c) od programiranja na simboličkom jeziku,
    d) od programiranja algoritama za rešavanje problema.
14. Koje su prednosti računarske mreže?
    a) brža dorada podataka,
    b) celishodnije korišćenje računarskih mogućnosti,
    c) manji troškovi za komunikacijske linije,
    d) pristup do bogatih baza podataka.
15. Čemu pripada operativni sistem: mašinskoj ili programskoj opremi računara?
16. Koji su zadaci operativnog sistema:
    a) priprema podataka za računar,
    b) rasterećivanje korisnika pri upotrebi računara,
    c) što bolje korišćenje mogućnosti računara,
    d) prevođenje programa.
17. Objasnite sledeće pojmove:
    a) baza podataka,
    b) ažuriranje podataka,
    c) zaštita podataka.
18. Šta je to računarska podrška informacionog sistema?
19. Šta je to upravljački informacioni sistem?
20. Koje su prednosti računarski vođene dokumentacije?
    a) pisanje dokumentacije nije potrebno,
    b) jednostavno unošenje izmena,
    c) smanjuje se obim posla,
    d) nije potrebno čitati dokumentaciju,
    e) dokumentacija je okupljena na jednom mestu,
    f) dokumentacija je ažurnija.

Odgovore objavljujemo na sledećoj strani

ODGOVORI

1 Računarstvo je nauka o svemu što je u vezi sa automatskom obradom podataka. Informatika je nauka o svemu što je u vezi sa obradom informacija odnosno podataka. Kibernetika je nauka koja se bavi procesima upravljanja u živim organizmima i mašinama
2 Algoritme koji su prilagođeni izvršavanju pomoću računara nazivamo program
3. Unos programa b unos podataka e
4 c
5 Numerička obrada a b, nenumerička obrada d, e, kombinovana obrada c, f
6 Diskretan zapis a c d neprekidan zapis b, e
7 Diskretan zapis b c neprekidan zapis a, d
8 Digitalni zapis povećavamo broj mesta u zapisu, analogni zapis povećavamo tačnost merenja fizičke veličine čije vrednost predstavlja zapis
9 Digitalni princip a, c d, analogni princip b
10 Prirodni jezici b e, veštački jezici a c d
11 b e f, h i
12 Prevodilac ume da prevede onu informaciju koja je zapisana na jeziku sa koga prevodi. Ako zapis nije u skladu sa gramatikom jezika prevodilac zaključuje da se radi o nekom drugom jeziku sa koga on ume da prevodi
13. b c d
14 b d
15. Deo programske opreme — softvera.
16. b c
17 a) Bogata banka podataka koji se odnose na isti problem i koji su na odgovarajući način povezani među sobom
b) Menjanje podataka da bi se podaci uskladili sa novonastalim stanjem
c) Obuhvata zaštitu podataka od neželjenih događaja kao što su uništavanje neodgovorno korišćenje menjanje i slično
18 To je sistem za prikupljanje, obradu, čuvanje i prikazivanje informacija pomoću računara
19 To je informacioni sistem koji na osnovu višeg nivoa obrade podataka — kao što je, na primer analiza međusobnog uticaja događaja praćenja promena pojedinih veličina u vremenu, predviđanja raznih događaja i slično — nudi podatke i analize potrebne za upravljanje
20 b, c, e f

## Rešenje testa iz prošlog broja

1 a) LIST
b) NEW
c) SAVE
d) LOAD
2. SAVE „PROG2"
LOAD „PROG1"
RUN
3 0________0
4 (grafikom ispušteno iz testa)
5 256
6 1024 bajta
7 2,4
8 2
9 2
10 S=3
11 2
12 10 P=A
20 A=B
30 B=P
13 X dobija vrednosti stepena 2 (2,4 8 16 ) dok ne dođe do prekoračenja i prekida izvršavanja programa
14 10 INPUT X Y
20 PRINT X+Y
30 END
15 110000
16 _12
_16
17 18
18. A=0
19 S=3/4
20 a) k>0, b) k>0 i k je parno
21 10 I=1
20 PRINT I
30 I=I+3
40 IF I=18 THEN 20
50 STOP
22 10 DIM A (1000)
20 FOR I=0 TO 1000
30 A(I)=1000−I
40 NEXT I
23 ABCABCABC
24 odgovor 3
25 usklađivanje sa novonastalim stanjem
b) zaštita od neovlašćenog pristupa i fizičkog oštećenja
26 OPEN CLOSE
27 naredbe aritmetičkih operacija, logičkih operacija, grananja prenosa,
28. odgovor 2
29 LIFO (Last In First Out) podatak koji se poslednji postavlja u stek prvi se uzima
30. izvršavanje aritmetičko-logičkih operacija i upravljanje radom ostalih komponenti sistema
31 6502 6510 (komodor 64 BBC) 8086/8088(IBM PC XT)

Takmičarski kutak

# Finale u aprilu

Od decembra prošle godine zadavali smo učenicima mlađim od šesnaest godine konkursne zadatke iz programiranja i one koji su poslali najbolja rešenja nagrađivali knjigama iz računarstva. Tokom pet meseci više od stotinu učenika osnovnih i srednjih škola iz čitave Jugoslavije slalo je svoja rešenja postavljenih zadataka koja smo pregledali i davali svoja zapažanja u rubrici „Pet plus". Rezultati ove akcije pokazali su da su konkursni zadaci podstakli i učenike iz najmanjih mesta, koji nemaju priliku da upoznaju računare u svojim školama, na programersko razmišljanje i bavljenje informatikom i računarstvom. Interesantna je činjenica da je većina učesnika ove akcije širenja računarske i algoritamske pismenosti iz malih mesta i više od toga oni su i među najboljima koji su učestvovali u finalnom takmičenju.

Ohrabreni uspehom prošlogodišnje akcije za otkrivanje mladih talentovanih programera od ovog broja „Računari" u saradnji sa Klubom mladih matematičara „Arhimedes" i Društvom matematičara SR Srbije ponovo pokreću ciklus konkursnih zadataka iz programiranja. Ove godine takmičenje će se obavljati u dve kategorije O — za učenike osnovnih škola i S — za učenike srednjih škola. Najbolja rešenja u obe kategorije nagrađivaćemo knjigama „Arhimedes" će svakog meseca nagraditi takmičara u mlađoj konkurenciji a Društvo matematičara najboljeg u starijoj konkurenciji. „Računari" će i ove godine bibliotekama škola iz kojih pristigne najviše rešenja, poklanjati komplete svojih izdanja. (Školama koje su u prethodnom ciklusu osvojile ukoričene komplete „Računara" izvinjavamo se što smo okasnili sa uručenjem nagrada i obećavamo da će ih dobiti pre kraja kalendarske godine.)

Učesnici koji u pet kola sakupe najviše poena biće u aprilu pozvani da učestvuju na našem finalnom takmičenju. Odatle će tri prvoplasirana u starijoj konkurenciji direktno ući na savezno takmičenje sa kojeg se bira ekipa za međunarodnu programersku olimpijadu. Svi ostali finalisti u obe konkurencije imaće pravo da idu direktno na odgovarajuće republičke takmičenje. Uz to pobednici u mlađoj konkurenciji dobiće besplatno letovanje sa „Arhimedesom" na Tari a pobednici u starijoj konkurenciji plaćena škola programiranja u Petnici.

## Evidencioni listić

Prezime i ime ______________________

Kategorija (O/S), razred i odeljenje ______________________

Škola i mesto ______________________

Kućna adresa i telefon ______________________

______________________

Broj poena (ispunjava Komisija) ______________________

## Konkursni zadaci — oktobar '88.

### a) Zadaci za učenike osnovnih škola

1 Sastaviti algoritam i napisati program za određivanje i štampanje svih parova prijateljskih brojeva manjih od 10000. (Dva prirodna broja m i n nazivaju se prijateljskim ako je zbir delilaca broja m jednak broju n i obrnuto zbir delilaca broja n jednak broju m. Na primer brojevi 220 i 284 su prijateljski jer za 220=2*2*5*11 i 284=2*2*71 važi 220=1+2+4+71+142 i 284=1+2+4+5+10+11+20+22+44+55+110)
2 Sastaviti algoritam i napisati program za računanje količnika dva cela broja tačno na zadati broj decimala

### b) Zadaci za učenike srednjih škola

1 Sastaviti algoritam i napisati program za računanje količnika dva cela broja tačno na zadati broj decimala
2 Sastaviti algoritam i napisati program za tačno računanje faktorijela prirodnog broja n manjeg od 200

Rešenja zadataka sa imenom rešavača slati na zasebnim papirima. Uz rešenja obavezno priložiti evidencioni listić (ili njegovu fotokopiju), koji nam je neophodan zbog ukupne evidencije o takmičarima i školama iz kojih se javljaju.

Rešenje svakog zadatka treba da ima algoritam listing i kratko objašnjenje programa, a poželjno je da ima i osvrt na moguća poboljšanja programa. Program može da bude napisan u bilo kom višem programskom jeziku za bilo koji personalni računar ali uz listing obavezno treba navesti jezik i verziju.

## Kriterijumi ocenjivanja rešenja

Rešenje svakog konkursnog zadatka može doneti učeniku od 1 do 5 bodova. Ispravna rešenja, u zavisnosti od toga koliko su dobra, mogu biti ocenjena

se tri, četiri ili pet bodova. Pod dobrim rešenjem podrazumeva se da je rešenje tačno, pregledno i korektno obrazloženo, da odgovarajući program radi što je moguće brže i zauzima što je manje moguće memorijskog prostora. Rešenje je bolje ako je dato tako da se može vršiti uopštavanje. Elegancija rešenja, mislimo pod tim na njegovu jednostavnost, takođe se uzima u obzir prilikom ocenjivanja. Ako se iz iste škole dobije više istovetnih rešenja, svi koji su ga poslali dobijaju po dva boda, jer nismo u mogućnosti da utvrdimo ko je u stvari autor takvog rešenja.

Rešenja treba da stignu u redakciju do 20. tekućeg meseca.

## Rešenje Prvulović Miloša za prvi konkursni zadatak „Tara 88"

```
10 INPUT „UNESI BROJ U BIN. ZAPISU";A$: A=0:N=LEN(A$)
20 FOR B=1 TO N: A=A+VAL(MID$(A$,B,1))*2↑(N-B): NEXT:REM KONVERZIJA
30 IF A=0 THEN A$=„NULA": GOTO 130
40 IF A>=1000 THEN PRINT „HILJADU";: A=A-1000
50 IF A<100 THEN 80 ELSE RESTORE 140
60 FOR B=1 TO INT(A/100): READ A$: NEXT
70 PRINT A$+„ ";:A=A-INT(A/100)*100
80 IF A<20 THEN 110 ELSE RESTORE 150
90 FOR B=2 TO INT(A/10): READ A$: NEXT
95 FOR B=2 TO INT(A/10): READ A$: NEXT
100 PRINT A$+„DESET";:A=A-INT(A/10)*10
110 RESTORE 160
120 FOR B=0 TO A:READ A$: NEXT
130 IF A>10 THEN PRINT A$+„NAEST" ELSE PRINT A$
140 DATA STO, DVESTA, TRISTA, ČETRISTO, PETSTO, ŠESTO, SEDAMSTO, OSAMSTO, DEVESTO
150 DATA DVA, TRI, ČETIRI, PET, ŠEST, SEDAM, OSAM, DEVET
160 DATA, JEDAN, DVA, TRI, ČETIRI, PET, ŠEST, SEDAM, OSAM, DEVET, DESET
170 DATA JEDAN, DVA, TRI, ČETIRI, PET, ŠEST, SEDAM, OSAM, DEVET
```

Program je pisan u GBASICu na računaru „tim 011".

U liniji 20 se vrši konverzija binarnog broja iz A$ u dekadni broj A. Zatim se pita da li je A nula, pa ako jeste, ispiše se „nula" i završi sa radom. Dalje se proverava da li je A>1000, pa ako jeste ispiše se „hiljadu" bez prelaska u novi red. Dalje se ispisuju stotine, desetice i jedinice. Program radi samo za brojeve (dekadno) do 1999 jer desetocifreni binarni broj nije veći od 1023. Program ne pravi nikakve nizove i koristi samo tri promenljive, od kojih je samo jedna string. Osnovna ideja je bila da se u DATA linijama ne nalaze cele reči, nego samo delovi koji su promenljivi. Tako za brojeve između 11 i 19 (uključivo) zajednički je nastavak „naest", za desetice „deset" i za stotine „sto", ali za stotine to nije iskorišćeno jer se 200 i 3000 izgovara „dvesta" i „trista". Program se može lako preraditi da radi i sa negativnim brojevima (ako je A<0 samo se ispred ispiše „minus") pa slovima A ili sa većim brojevima, jer se čitanje broja svodi na čitanje klase kao da je broj do hiljadu a zatim se ispiše osnova klase („milion", „hiljadu"). Naravno, mora se paziti na rod i broj (dve hiljade, dva miliona i slično).

## Rešenje Lazić Renka za drugi konkursni zadatak

```
10 INPUT X: DIM A(X,2): LET M=
100: LET X1=0: LET Y1=0: LET X2=
0: LET Y=00
20 FOR I=1 TO X: INPUT A(I,1),
A (I,2): PLOT A(I,1), A(I,2)
30 NEXT I
40 FOR I=1 TO X-1
50 FOR J=I+1 TO X
60 DEF FN A(X)=(A(I,2)-A(J,2))
/(A(I,1)-A(J,1))*(X-A(I,1))+A(I,
2): LET S=0: LET N=0
70 FOR K=1 TO X
75 IF (A(K,1)=A(I,1) AND A(K,2
)=A(I,2)) OR (A(K,1)=A(J,1) AND
A(K,2)=A(J,2)) THEN GO TO 110
77 IF A(I,1)=A(J,1) THEN LET U
=A(K,2)+SGN (A(K,1)-A(I,1)): GO TO 90
80 LET U=FN A(A(K,1))
90 IF U>A(K,2) THEN LET S=S+1
100 IF U<A(K,2) THEN LET N=N+1
110 NEXT K
120 LET P=ABS (X-N-2-2*S): IF P
<M THEN LET M=P: LET X1=A(I,1):
LET Y1=A(I,2): LET X2=A(J,1): LE
T Y2=A(J,2)
130 NEXT J: NEXT I
140 PLOT X1,Y1
150 DRAW X2-X1,Y2-Y1
160 PRINT „PRAVA JE ODREĐENA T
AČKAMA ";"(";X1;",";Y1;")(";X2;
",";Y2;")"
```

Program je pisan u bejziku računara „spektrum".

Prvi deo 10—30 se bavi unosom podataka. A(x,2) sadrži x i y koordinate tačaka, m sadrži zasada minimalnu razliku tačaka sa raznih strana prave, x1 y1 x2 y2 sadrže koordinate tačaka kojima je određena prava sa čijih strana se nalazi sada minimalna razlika tačaka. Petlja 40—130 je simulacija buble sorta, tako što bira parove tačaka (svaka sa svakom) i kroz njih provlači pravu. Redovi 75—100 prebrojavaju tačke sa jedne strane prave i tačke na pravoj. Broj tačaka sa druge strane prave se izračunava u liniji 120 na osnovu prethodnih podataka. Tu se ujedno proverava da li je razlika broja tačaka sa raznih strana prave manja od m, pa ako jeste — postavlja nove vrednosti promenljivima m, x1, y1 x2 i y2. Linija 77 se brine da ne dođe do deljenja sa 0 a linija 75 se stara da se preskače provera tačaka kojima je prava određena. 140—160 linije zadužene su za crtanje prave i ispisivanje koordinata kojima je određena.

## Rešenje Prvulović Miloša za treći konkursni zadatak

```
10 INPUT „DIMENZIJE";M, N:DIM L(M,N)
20 FOR B=1 TO N: FOR A=1 TO M
30 INPUT L(A,B): NEXT A,B
40 FOR B=1 TO N: FOR A=1 TO M
50 IF L(A,B)=0 THEN GOSUB 160:GOTO 120
60 IF B=1 THEN 0=1: GOTO 100
70 IF L(A,B-1)<>2 THEN 0=1
80 IF 0=0 THEN L(A,B)=2
90 IF B=N THEN 120
100 IF L(A,B+1)=1 THEN P=P+1
110 IF P>1 THEN L(A,B)=2
120 Z=1
130 NEXT: GOSUB 150: NEXT
140 PRINT „CEPA SE NA";S;„DELOVA": END
150 IF P=0 AND Z=1 AND 0=1 THEN S=S+1 ELSE P=0
160 0=0: Z=0: RETURN
```

Program je pisan u GBASICu za računar „tim 011".

Program radi na sledeći način. Vrši se prolaz kroz matricu (0 je rupa, 1 je papir) i traže blokovi papira u horizontalnim redovima. Kada se posle rupe ili početka reda naiđe na papir, proverava se da li je ispod tog kvadratića takođe papir. Ako nije, ide se dalje na sledeći kvadratić. Ako se naiđe na rupu ili kraj reda, a ispod reda papira nije bilo ni jednog papirnog kvadratića, onda se znači završio komad papira, pa se on prebroji sa S=S+1. Ako se ispod reda papira naiđe ponovo na papir, onda se taj red ne broji, jer će se prebrojati onaj papir ispod njega ili eventualno onaj koji se nalazi još niže. Ako se nađe više kvadratića papira ispod jednog bloka papira, samo iznad prvog se stavlja jedan, a iznad ostalih 2 da bi, kada se bude pretraživao sledeći red, računar „znao" da su svi ti eventualno različiti blokovi papira u stvari samo delovi jedinstvenog bloka papira. To se postiže tako što se blok ne broji ako se iznad njega ave dvojke bez jedne jedinice.

Program možda izgleda komplikovano, ali samo zbog nestrukturiranosti ove verzije bejzika. Jedna IF ... THEN ... ELSE struktura bi zamenila sve linije u sredini programa. Osnovna prednost ovako zamišljenog pretraživanja je kratkoća i brzina izvršavanja programa, jer se obavlja tačno jedan prolaz kroz matricu i ni jednom elementu matrice se ne pristupa dva puta.

## Rešenje Lazić Renka za četvrti konkursni zadatak

```
10 INPUT N,M: DIM A(N,M)
20 FOR I=1 TO N: FOR J=1 TO M
30 INPUT A(I,J)
40 PRINT AT J-1,I-1,A(I,J)
50 NEXT J: NEXT I
60 LET X1=1: LET Y1=1
70 LET X2=3: LET Y2=1
80 IF A(X1,Y1)+A(X1+1,Y1)+A(X1,
Y1+1)+A(X1+1,Y1+1)>A(X2,Y1)+A(X
2+1,Y2)+A(X2,Y2+1)+A(X2+1,Y2+1)
THEN FOR K=1 TO 4: READ Q, W, E, R:
LET Z=A(Q,W): LET A(Q,W)=A(E,R):
LET A(E,R)=Z: NEXT K: RESTORE
90 IF X1=N-3 AND Y1=M-1 THEN GO TO 120
100 IF X2=N-1 AND Y2=M-1 THEN GO
TO 150
110 IF X2=N-1 THEN LET X2=1: LE
T Y2=Y2+2: GO TO 80: ELSE LET X2
=X2+2: GO TO 80
120 FOR I=1 TO N: FOR J=1 TO M
130 PRINT AT J+M,I-1,A(I,J)
140 NEXT J: NEXT I: STOP
150 IF X1=N-1 THEN LET X1=1: LE
T Y1=Y1+2 ELSE LET X1=X1+2
160 IF X1=N-1 THEN LET X2=1: LE
T Y2=Y1+2: ELSE LET X2=X1+2: LET
Y2=Y1
170 GO TO 80
180 DATA X1,Y1,X2,Y2,X1+1,Y1,X2
+1,Y2,X1,Y1+1,X2,Y2+1,X1+1,Y1+
1,X2,Y2+1
```

Program je pisan u bejziku računara „spektrum".

Linije 10—50 predstavljaju pripremu programa. A(n,m) je matrica. U linijama 60 i 70 se postavljaju promenljive za buble sort. X1 i y1 su indeksi koji odgovaraju spoljnoj petlji sortiranja, a x2 i y2 unutrašnjoj. U liniji 80 se upoređuju zbirovi podmatrica. Ako je prva veća od druge vrši se zamena elemenata. Linije 90—120 i 150—180 se brinu o kontroli toka. One proveravaju da li je unutrašnja petlja stigla do kraja da li je neka petlja stigla do ivice i sl. U linijama 120—140 se ispisuje novodobijena matrica.

Rešenja petog i šestog konkursnog zadatka „Tara '88" objavićemo u narednom broju.

### Veseli odmor

- Programeri ne stare — zastarevaju njihovi programi.
  - Najbolji način za očuvanje programerskog mentalnog zdravlja je da im se potvrdi kako je za sve kriva mašina.
- Manja fizička nasilja nad računarima ne moraju obavezno biti pogubna po njihovu dušu, a sigurno opuštaju programere.
  - Svaki novi program po pravilu predstavlja dobro zaboravljen stari.
- Upamtite da je mnogo lakše ući u program nego iz njega izaći.
  - Vodite računa da uloženi ciklusi ne treba da škripe pri radu.
- Osnovna dilema programiranja je šta je komplikovanije: program ili programer koji ga je napisao?
  - Korišćenje disk jedinice kad stočića za ručavanje može jako da iznervira, jer dotična razdražujuća vibrira.
- Zabranjena je upotreba aspirina, kiselog mleka i alkohola za skidanje temperature pregrejanom računaru.

Razbarušeni sprajtovi



*Uređuje Slobodan Perović*

# GALACTIC GAMES

Kada sam učitao ovu igru pojavilo se oko koja je ispisivalo uputstva. Već tada sam znao da imam posla sa vrlo dobro urađenom i vrlo duhovitom igrom. Kasnije sam doznao da se je oko zove Harry i da izvještava za Galaktoviziju. Harry će vas pratiti tokom cijele igre, obavještavajući vas o novoj disciplini (i opisujući takmičare) ta vam pokušati olakšati poraz duhovitim komentarom o prolaskoj igri. Kada učitate svaku disciplinu pojavit će se njeno stilizirano ime dok prolazi kuz svamir.
Da krenamo redom.

1. 100 m SLIGHTER (puzanje)
Ekran je podijeljen na 3 dijela. U gornjem dijelu su mjerači TEMP — temperatura i SLIME — snaga. U srednjem djelu je kompjuterski gmizač, a u donjem ste vi. Postoja dva načina da se krećete: bacaju ći se (trebate imati dovoljno snaga) i puze ći. Utrka je dosta smiješna. Kada kompjuter uđe u cilj stanja i smiješći se podigne dok vi ostanete ležati iscrpljeni i veoma jadna izgleda. Najbolja vam je taktika da se jednom na početku bacite pa nastavite puzati dok vam mjerač temperature (TEMP) na pređe polovicu. Zatim se opet bacite (dole pa desno) pa puzite (dalje pa gore) dok vam TEMP na pocrvani. Sa će sva do cilja pokušavajte se baciti naprijed.

2. SPACE HOCKEY (svemirski hokej)
U ovoj disciplini se transformirate u biće koje živi u istom sunčevom sistemu kao i Harry (valjda su zato toliko slični). Vaš cilj je da odbijajuć pak sa planete PRCZYX pogodite protivnički gol. Samo pazite pak je pomalo svojeglav pa mu se dasi da ekrana sa putanje kada se toma budete najmanja nadali. Također je neobično i to da su golovi veoma gladni jer vam se (ako se suviše približite) može desiti da budete prosto usrkani od strane svog ili protivničkog gola. U tom slučaju protivnik dobija 1 poen (za ubacivanje paka dobija se 3 poena). Utakmica traje 4 četvrtine po 60 sekundi. Iako grafika nije našto na ročito vjerujam da ćete se s ovom disciplinom mnogo zabaviti.

3. PSYCHIC JUDO (psiho džudo)
U ovoj disciplini, koja se odlikuje dobrom grafikom, postoje 2 pokazatelja energije (koju gubite ako vas neprijatelj pogodi) i snaga udara (povećava se pritiskom na FIRE). Na ekranu su nacrtana dva lika, koji jedan na drugog iz stomaka ispaljuju vatrena kugle. Putanju kugle možete mijenjati za vrijeme čitavog njenog leta, a na kompjuterov napad morate odgovoriti bra njenjem dijela tijela za koji pretpostavljate da će biti na udaru kugle. Na preporučujem vam da puno promašujete jer zavisno o jačini udarca (određujete je pritiskom na FIRE) gubite dio energije. Meč traje dok neko od džudista ne izgubi energiju ili 60 sekundi. Ukupni pobjednik je onaj ko dobije 3 meča.

4. HEAD THROWING (bacanje glava)
Ovo je svakako najsmješnija disciplina. Grafika i animacija su kao u crtanom filmu. Cilj je baciti glavu što više udalj (tu glupi imaju sraću jer im je glava lakša pa više leti). Prva stvar je hvatanja zaleta — tu vam pomaža iskustvo iz Decathlona i Supertesta. Kad dođete do crne crte pritisnite Fire i ne otpuštajte ga dok glava koja se pojavila u kvadratu na vrhu ekrana na dođe u željani položaj za izbacivanja. Zatim otpustite Fire i ne gledajući na ekran prilično lipku za gore. Sada vaša glava maše ušima što joj produžava let. Otprilike 3 sekunda prije kraja leta pritisnite još jednom FIRE i glava će se zabiti nosom u zemlju. Na učinila li to bacanja će biti neregularno. Nakoliko trenutaka nakon što je završila s nosom u zemlji odletjet će sa ekrana, a za njom će trčati vaš bezglavi takmičar pokušavajući da je dohvati.

5. METAMORPH MARATON (transformirajući maraton)
Na kraju najteža discipline. Grafički nije našto naročito ali će vas dugo dugo držati uz kompjuter. Upravljate sa 4 lika: trkačem, skakačem, helikopterom i likom u obliku zvona za koji još na znemo za što služi. Pritiskom na FIRE dobijate mogućnost mijenjanja likova u toku maratona.

Na vrhu ekrana nalaze se vrijeme koje je proteklo od starta, vaša energija koja se vremenom smanjuje (kad nje nestane — maraton is over), te mjerač za snagu vašeg skoka ili brzinu trčanja. Uz napomenu da na putu nalazite kapsule koje vam obnavljaju energiju, ostavljamo vam da sami nađete put do cilja.

Općenito uzevši ovo je jedna od boljih igara sa sufiksom „games" u kojoj je prisutna velika doza humora. Dakle ukoliko volite igre ovakvog tipa, nabavite GG a ako ih na volite zavalit ćete ih nabavlja jući ovu igru.

*Dražen Klasić, Split*

## Usijan džojstik

Sa molbom da pritekneme u ... imam igru 'Tai pan' na kaseti ... U principu sve bi trebalo da se radi ... „speed" ... predviđam ali sve zavisi od toga ... varzja isp...

* * *

... i *Nino Pešak* iz Zagreba mole da im izađemo u susret za igru „Death wish II".
Vi ste na potezu.

* * *

Igrala ir igru „Captain Amerika"?
Možda će vam pomoći savet *Kristijena Kaurića* iz Buja:
„Ako izgubite šit dođite do komandne table ... „Q-SHIFT CRSR LIJEVO I DESNO" i opet ste na konju."
Ovo važi za C 64.

* * *

... pitanja za vas: Kako u igri „Eden blues" izaći iz prve sobe?
... očekuje *Ratko Popević* iz Požarevca sa stanom u ulici 7 ... broj je 12000.

* * *

... a rekordima.
... nam *Jovan Gavović* ali na žalost ne znamo ... jer je ... nečitak.
Moj ortak Miroš već vam se javio da je rasturio „Uridium" i nakucao (milion i šest stotina hiljada kusur) a ja sam to uradio u igri „Terra ... sam nakupio 700050 poena!"
Kako se to i ... 

* * *

*Andrej Mrzel* iz Ševa pri Ulji, *Dejan Milanković* iz Subotice i *Vladimir ...* iz Beograda obratili su nam se sa identičnim zahtevom. Nude razne opise ... da izaberemo one koje ćemo objaviti.
U principu mi smo za sve što vec nije objavljeno ali sve zavisi od kvaliteta. U svakom slučaju bolje je poslati tekst nego čekati što posebno važi za ... Pavlovića, koji je sa nama veoma uspešno sarađivao.
Takođe, broj priloga u „razbarušenim sprajtovima" nije ograničen pa su ... svaki treba biti uspešan, zanimljiv i raznovrstan.
Evo i predloga koji nam je stigao od *Jasmina Klarića* iz Splita. Želja mu je da svi zajedno rešavamo jednu tešku igru koja nosi ima „Dark Sceptre". Igru još ... — nastojaćemo da je nabavimo — a u međuvremenu prostor u „Razbarušenim sprajtovima" je na raspolaganju svima koji žele da se uključe.

* * *

Prilika je da pomognemo i Draženu Klarću iz Splita. Da bi seo u ... „School daze" priđi žaljenom mestu ... pritisni tastar S.

* * *

Evo i ... nepopularna lista onih koji su zakasnili sa opisima: *Mario P...*, Dejan Nikolić iz Niša, *Ivan Pavlović* iz Beograda, *Predrag Stanisavljević* iz Smederevа, *Danijel Mihajlović* iz Tuzle i *Dejan Fahirov* iz Bečeja.
Više sreće drugi put.

* * *

*Predrag Zerdo* iz Dubrovnika poslao nam je kratak opis igre „Sexy Black Jack" ... između ostalog piše ovo:
„... igrate protiv jedne dame koju morate četiri puta zaredom pobediti da biste videli njene čari."
Daleko od toga da imamo nešto protiv ali zar nije uputnije dok je sunce i ... malo zapostaviti računar i baviti se nečim ... 

* * *

I na kraju podsećamo vas da rubriku „Razbarušeni sprajtovi" realizujemo ... sa emisijom „Čip i sedam ... " koja se svake subote počev od 14. februara emituje na Prvom programu Radio Beograda.
Naš zajednički telefon je (011) 339-070. Očekujemo da nam se javite ...

# SOLOMON'S KEY

**„amstrad"**

Vi ste jedan mali čovek i tražite izlaz iz sobe u kojoj se trenutno nalazite. Zašto uopšte nije važno. U prostoriji se krećete tako da izgradite put kamenjem kojeg pravite pritiskom na SPACE. Ostale komande su:

A — dole (čučanj)
Q — gore
O — levo
P — desno
ESC — pauza (FIRE — ponovna igra)
2×ESC — prekid
ENTER/RETURN — ispaljivanje vatrenih kugli

Kamenje možete postavljati u različitim pravcima sa kombinacijama SPACE, SPACE+(A, Q). Naravno možete i uništavati kamenje koje se nalazi u sobi i to radite sa istim komandama. Uzput skupljajte različite stvari koje vam donose bonus poene ili neka poboljšanja. Obratite pažnju na LISTINU (odmah prelazite u sledeću sobu) i na CRVENO-ŽUTE BOCE (novi život). Izlaz otvorite sa KLJUČEM. ZVONCE oslobađa djevojku koju možete uhvatiti ako volite žene ili neva poena.

A kakva bi igra bila ako ne bi bilo neprijatelja? To su razne životinje, vatre, zombiji! Sve ove stvorove ćete lako nadmudriti ili vatrenom kuglom ubiti ako im dobro proučite kretanje. Vatrene kugle se nalaze skrivene u PLAVIM BOCAMA. Biće vam od velike koristi!

A sada pokušajte preći tih 20 ekrana praćenih odličnom muzikom i lepom grafikom.

Upozorenje: igra sa poukom nije više zanimljiva!!

*Miha Logar, Kranj*

# XENON

**„amiga"/„atari ST"**

Xenon je jedna od najnovijih i najboljih igara za „amigu" i „atari ST" u poslednje vreme. Priča koja prethodi akciji je bezbroj puta prežvakana. Pomoću lasera i svoje hitrosti treba se probiti do kapetana Xoda koga su zarobili zli Xeniti. Sva ostalo je cela pucnjava, a svoj brod po već ustaljenom receptu možete opremiti i raznim dodatnim oružjem koje sakupljate duž puta tako što prelazite preko kvadratića sa određenim slovima.

Pored toga postoji mogućnost da po potrebi vaš brod pretvorite u neku vrstu tenka kojim se možete kretati po površini planete i samim tim uništavati ciljeve na Zemlji — međutim to i nije baš preporučljivo.

Sve u svemu postoji šesnaest nivoa. Posle pređena trećina puta na svakom nivou vas čeka jedno čudovište koje liči na veliku svemirsku krstaricu veće od vašeg broda pet do šest puta. Da bi ste ga uništili i prošli dalje morate ga nekoliko puta pogoditi u središnu narandžastu kupolu. Isti neprijatelj vas čeka i na krajevima nivoa.

Inače, pored osnovnog oružja — prednjeg lasera — postoje i bočni ili dijagonalni laseri, bacači plamena itd.

Iako je ovo ipak jedna od onih igara koje i ne pomišljamo da završimo, mislim da je do sada u svom žanru i najbolja!

*Sveta Petrović, Beograd*

782

# SOCCER BOSS

Ovo je još jedan u nizu programa za C-64 koji vas postavlja na mesto trenera fudbalskog tima. Pošto svaki Jugosloven zna sve o fudbalu, evo prilika da se dokaže!

Odmah po učitavanju postavlja se pitanje da li ćete da učitate raniju igranu igru. Obratite pažnju na ovo jer je to prva i poslednja šansa da ja učitate. Posle toga sledi izbor imena koji ćete voditi. Ako ne izaberete neki od 60 klubova iz 4 engleske lige, naimenovačete novi. Koji god tim da uzmete na početku ćete imati iste igrače i isti konto u banci — 500.000 funti.

Posle toga sledi „osnovni" skrin sa svim gore je naziv vašeg kluba levo od njega je broj liga u kojoj se trenutno takmičite a desno je broj nedelja. Zatim sledi spisak igrača i razervi. Igrača opisuje samo dve stvari redni broj u ekipi i kvalitet. Kvalitet može imati vrednost od Ø—9 i što je veći igrač je bolji (vrlo logično!). Pored toga s igračeve leve strane može da stoji plus ili minus. Plus znači da je igrač u formi i da će mu kvalitet posle sledeće utakmice (najverovatnije) porasti, minus znači suprotno.

Igraču nije određeno koje mesto u timu igra (odbrana, srednji red ili napad) tako da igrač može igrati praktično svugde. Izdvojeni su jedino golmani koji imaju „tarabu" (SHIFT i 3) ispred svoga imena.

Posle toga dolazi količina novca u banci broj sezona, sledeći protivnik i njegov rejting. Ispod se nalazi formacija u kojoj igra vaša ekipa, a desno od nje je rejting vašeg tima. Rejting se dobija tako što se saberu vrednosti igrača koji igraju u odbrani (srednjem redu, napadu) i podeli se brojem igrača koji igraju u njemu.

Konačno dolazimo do menija kojim kontrolišete i menjate sva do sada opisane.

— Formation (formacija poziva se na F) — ovim se menja formacija, to jest broj igrača koji igraju u odbrani, srednjem redu i napadu

— Move (premeštanje, M) — premeštanje igrača sa jednog mesta na drugo ubacivanje u igru, slanje na klupu za rezerve. Pred svaku utakmicu dobro je proveriti rejting jer se može desiti da, na primer rejting odbrana bude 16 4=4. Ako sada neki igrač iz odbrane ima vrednost 7 a neki igrač iz napada 9 zamenivši im mesta pojačaćete napad bez kupovine igrača i maksimalno iskoristiti trenutnu ekipu

— Chek (provera, C) — ovim se dolazi u obiman meni koji sadrži

1. ON THE BANCH prikazuje rezerve i igrače koji su „van akcije" — to su suspendovani ili povređeni igrači

2. TRANSFER MARKET vam nudi izbor od 15 što igrača što golmana. Da biste mogli da kupite nešto potrebno je da imate novaca i mesta u ekipi. Cena igrača se dobija tako što pomnožite vrednost igrača sa 10.000 funti. Lista je praktično neiscrpna jer kad kupite igrača, na njegovo mesto uskače drugi.

3. LEAGUE TABLE tabela vaše lige

4. FIXTURES ovde se vidi raspored timova za kup

5. BANK MANAGER pošto utakmice ne možeš igrati dok si u „minusu", njega možeš nadoknaditi prodajom igrača ili pozajmicom od banke. Limit za pozajmljivanje postoji. Banka sedmično (posle svake utakmice) uzima sa tvog računa 0,5% interesa i 5% rate na sumu koju si pozajmio. To je otplata, tako da se dug vremenom smanjuje a time i interes i rate. Kad skupiš dovoljno novaca za otplatu duga, banka će ga podići. Ovde se nalazi još i limit do kojeg možeš biti u „minusu" (overdraft limit). Granice se pomeraju u odnosu na to u kojoj ligi igraš

## POUKOVA MREŽA

„Spektrum" — *Boban Ilić*

| | | EFEKAT |
|---|---|---|
| WIZ | 60306,0 | Energija |
| STREET HABLE | 40661,36 | Životi |
| BUGGY BOY | 39066,0 | Vreme |
| XARAX | 41352,0 | Životi |
| VIXEN 1,2,3 | 65519,38 | Životi |
| | 65519,201 | Energija |
| Hundra | 40716,0 | Životi |
| | 41375,0 | Životi |

6—9 DIVISION 1—4 SUMMARY ovu opciju koristite kada želite da pogledate ko zauzima prva tri ili zadnje tri mesta u ligama

◊ LEADING SCORERS ovde je lista najboljih strelaca. Na žalost odnosi se samo na vaš klub a kazuje koga treba staviti u tim a koga prodati

♦ SAVE snimanje trenutnog statusa. Status je najbolje snimiti na kraju sezone kada se plasirate u viši rang jer su igrači tada u najboljoj formi

— Sell (prodaja S) — ovom naredbom prodajete igrača. Za njih se dobija otpnilka 7500 funti po jedinici vrednosti. Ova naredba omogućava i bavljenje „malom privredom" na Transfer marketu uzmite igrača koma je vrednost Ø sačekajte da mu poraste pa ga predajte!

— Play (igranje P) — kada si zadovoljan ekipom možeš početi susret.

Prilikom utakmice u gornjoj polovini ekrana nacrtan je šesnaesterac. U njemu je broj nedeljno rezultat i sat. Iznad njega je napisano s kim igrate a ispod njega je prostor u koma se ispisuje koja je ekipa dala gol.

Nakon utakmice sleda rezultati susreta a zatim tabela. Tabela je kao i svaka druga sem što se za pobedu dobija tri boda a ne dve kao kod nas. Tako da će pri kraju prvenstva nekoliko klubova „iskočiti" od ostalih zbog nedostižnog broja bodova. Posle tabele sledi obavezni prihod od ulaznica i troškovi plata, a posle toga nasumice odabrani troškovi i prihodi.

Grafika u ovom programu skoro da i nema a zvuka je još manje. Ipak igra ima „atmosferu" koja će biti u stanju da vas zadrži i duže od pola dana! Osim toga, u nižim ligama lako ćete se probiti do prvog mesta. A u prvoj?

Srđan Gligorić, Bečej

# SORCERER LORD

SORCERER LORD je strateška simulacija. U ulozi ste lorda Galancra, čiji je zadatak da zaustavi agresiju zločestog Schatten lorda i zauzme njegove zamkove (FESTUNG VON SCHATTEN LORD). U tome će vam pomoći LORD VON ROVANIUM, LORD VON HEREDACH, LORD VON MORGALION, LORD VON SAVANTRON i LORD VON IMRYTH. Svaki od njih raspolaže vojskom koju čine pešaci (KRIEGER) i konjanici (REITHER). Pored toga poseduju zamkove iz kojih, pritiskom na taster G, mogu regrutovati (taster 1) ili postaviti (taster 2) stražu. U toku igre vojska se obnavlja po zamkovima. Komande za pomeranje kursora po ekranu su E — severozapad, R — severoistok, X — jugozapad, C — jugoistok, S — zapad i F — istok. Tasterom O birate hoće li se jedinica kretati (1) normalnim ili (2) forsiranim maršom. Pri tom vodite računa da se preko planine i šume jedinica može kretati samo forsiranim maršom. Kod takvog kretanja pazite da vam se vojska ne premori jer neće moći da nastavi kretanje a u bici će trpeti velike gubitke. Pritiskom na M dobijate mapu predela, T vam služi za podatke na čijoj ste teritoriji, a pritiskom na O završavate potez.

Posle svakog odigranog poteza možete birati da li ćete 1 napustiti igru 2 učitati poziciju 3 snimiti poziciju ili 4 nastaviti igru. Između poteza dobijate podatke o broju poginulih (sa obe strane) s lordovima koji su vam verni i broju prstenova (RUNNEN RING) koji se nalaze kod vas i Schatten lorda. Ovi prstenovi su veoma važni za dalji tok igre. Naime ako izgubite sva prstenova igra se završava. U početku ih meta osam a Schatten lord nijedan. U slučaju da vam Schatten lord oduzme prsten ne očajavajte jer ga možete povratiti. Dovoljno je se bilo kojim lordom doći do njega i uzeti ga. To bi bilo sva o komandama u igri. A sad ko misli da je dobar strateg neka ne čita sledeće redove. Oni su namenjeni onima koji ni posle više puta nisu uspeli da savladaju Schatten lorda. Što pre regrutujte sve lordove i rasporedite ih uz RHARTA VON SAVANTOR? ELVIOS VON IMRYTH I CHELBYTH VON IMRYTH. Ova zamkova Schatten lord uvek prvo napadne i važna su strategijska tačke. Izaberite bilo kog lorda sa kojim ćete otići u HOCHBURG VON SITHRIN i HOCHBURG VON YARTHROS. U ovim zamkovima vojska se veoma brzo obnavlja. Tu regrutujte vojsku i sa njom krenite u pomoć opsednutim zamkovima. Sačekajte da Schatten lord izgubi svu vojsku u napadima na vaše zamkove i zatim sa ostatkom vojske krenite u zauzimanja njegovih zamkova. Pri tome pazite da na prođe više od 50 dana od početka igra jer se ona u protivnom završava. Šta još reći o SORCERER LORD-u? Grafika je lepo urađena, zvuk se javlja samo u nekoliko zvižduka, a koncepcija igre neodoljivo podseća na LORDS OF MIDNIGHT!

Slobodan Miljković, Beograd

## POUKOVA MREŽA

„Komodor" — *Ivan Jelčić*

| | | |
|---|---|---|
| HUMAN RACE | 8136,165 | Životi |
| SIDE WIZE | 23736,111 | Životi |
| JET BOYS | 3261,173 | Životi |
| PIR 2 | 3627,98 | Životi |
| PREDATOR | 34862,156 | Životi |
| TARGET RENEGADE | 43551,45 | Besmrtnost |
| | 43261,45 | Vreme |

# BASKET MASTER

## „Spektrum" 48/„Komodor" 2

Na „Spektrumu" je ovo sigurno najbolja košarka, dok je na „Komodoru" samo još jedna prosječna simulacija košarke. Igra se odvija na dva koša. Broj udaraca je stvarno veliki. Na raspolaganju su skoro svi udarci uključujući i slobodna bacanja. Teren je dobro obilježen na „spektrumu" dok na „komodoru" nije obilježena linija 3 poena, nego se svaki postignuti koš izvan reketa računa kao 3 poena. Kod „komodora" još zbunjuje i to što loptu ne možete pucati iz svoje polovine terena. Takođe je nemoguće dati koš iz daljine nešto veće od dužine reketa. Kod „spektruma" možete dati koš i preko cijelog terena. Meni je kod „komodoru" veoma skučen. Sastoji se iz 4 opcije

1. Igrač protiv kompjutera
2. Dva igrača
3. Nivo umijeća kompjutera (početnik) amater profesionalac (NBA) i
4. Mijenjanje imena (kompjuteru ne možete promijeniti ime)

Kod „spektruma" još postoji izbor raznih palica, tastatura, a možete odrediti i vlastite tastere. Zatim počinje utakmica. Igra se u dvorani sa publikom koja bučno pozdravlja vaše uspješne poteze. Gornji dio ekrana je rezervisan za samu igru. U donjem dijelu se nalazi sat koji odbrojava vrijeme i podaci o igračima. Igračeva energija (što je veća, igrač je brži i precizniji) broj ličnih grešaka (posle 5 grešaka utakmica se završava), broj poena, u čijem je posjedu lopta, ime igrača i mjesto za poruke (koš, faul, poskok, aut itd). Na početku igrači kreću sa krajeva terena, a lopta se izbacuje sa stalona. Igraju se dva poluvremena po 5 minuta a na kraju svakog poluvremena se izbacuju statistički podaci za svakog igrača postotak šuta za 1,2 ili 3 poena, broj skokova lične greške. Igre posjeduje mnogo šuteva normalan šut se izvodi kad igrač stoji okrenut prema košu ili mu je iza leđa, pa se automatski okrene, horog kad stoji paralelno sa košem ili mu je okrenut leđima (samo u nekim slučajevima) zakucavanja se izvodi ovako dođete ispod samog koša i dva puta brzo pucate. Možete se okrenuti u zraku i zakucati s leđa. Svako zakucavanje biće propraćeno ponovljenim snimkom ali sa uvećanim likovima. Ako prilikom slobodnih bacanja uključite automatsko pucanje lopta će se zavrtiti i ući u koš. Ovo se može izvesti samo na „komodoru"

Inače, slobodna bacanja se izvode na principu jedan za jedan. Kao i u ostalim simulacijama košarke, šut se izvodi tako da sa prvim pritiskom na pucanje skočite, a zatim drugim pritiskom pucate. Možete skakati ispod koševa, uhvatiti loptu i u letu je poslati u koš. Pravac kretanja određujete sa smjer+pucanje. Protivnika ćete najlakše izazvati da napravi ličnu ako mu okrenete leđa i gurate ga. Svaki njegov pokušaj oduzimanja lopte sa pucanjem će mu pridodati još jednu ličnu grešku. Lična greška je i ako igrač pokuša oduzeti loptu protivniku dok je ovaj u skoku. Evo kako ćete najlakše pobijediti kompjuter

Na „komodoru" namjamo napravite aut i kad kompjuter pođe sa svog dijela terena pritrčite mu oduzmite mu loptu i pošaljite je u koš. Zaobiđite protivnika i ako ste promašili zakucajte loptu

Kod „spektruma" sačekajte da vam protivnik priđe i izgubi dio energije pucajte loptu, zaobiđite protivnika (pošto imate više energije) i ako ste promašili prvi put sad dovršite posao. Često se okrećite. Utakmicu dobijate tricama koje nije teško postići.

*Mario Miličević*

## Prikaz meseca

*Nagrada je pripala Zoranu Jovanoviću iz Niša za prikaz igre „Flintstones". Pored toga, u toku septembra meseca u emisiji „Čip i sedam jarića" objavljeni su tekstovi Srđana Gligorića („Winnie The Pooh"), Zlatana Hamzića („The Race against Time for Sport Aid 88") i Vladimira Jankovića („Dark Side")*

*Gotovo je gotovo*

# THE FLINTSTONES

*Praistorijsko kameno doba, gde se radnja ove igre dešava, ne nudi obilje zabave, pa je Fredu glavni hobi partija kuglanja sa najboljim prijateljem Barnijem. Međutim, ovoga puta Fred neće moći na uobičajeno kuglanje, jer mu je Vilma naredila da okreči većinu u kojoj žive (naravno, tada nisu postojali učebni sta veni i moleri) i da budno motri na njihovog devojčurka Pebbles, dok ona bude u kupovini sa svojom prijateljicom Beti. Krečenje nije veliki problem, ali tu je nevaljala devojčica Pebbles, koja stalno šara po zidovima i pravi Fredu probleme pri krečenju. Zato Fred mora stalno da je vraća u kolevku, potom da bukvalno jori krpu kojom kreči, i da naravno kad god mu ponestane farbe krpu natopi u sud sa farbom. Na raspolaganju su i merdevine, koje možete da pomerate i sebi olakšate krečenje. Treba reći da su veoma duhoviti Vilmini komentari (i Fredovi također) kada u predviđeno vreme ne uspete da okrečite ceo zid. Ukoliko ipak uspete da završite kućni posao, Vilma dozvoljava Fredu da pođe na kuglanje. Znači, prelazite na drugi nivo.*

*Nalazite se u praistorijskom vozilu i čeka vas vožnja vrlo lošim putem do kuglane. Put je neravan i pun rupa, koje morate preskakati, tako da od vaše spretnosti zavisi da li ćete uspeti da pređete ovaj nivo i stignete na treći u kuglanu, koji učitavate sa kasete. Napokon vas čeka kuglanje sa Barnijem. Sve ovo je već viđeno, tako da mi je kuglanje bilo dosadno (između ostalog i zbog toga što je lako pobediti Barnija). Kada mislite da je sve završeno, dolazi do obaveštenja da je Pebbles u vašem odsustvu napravila lom u pećini i nestala. Čeka vas poslednji platformski nivo, u kome se morate probiti do svoje ćerkice (vrlo nevaljale ćerkice). Ona se nalazi gore levo i ... tu je kraj.*

*Zoran Jovanović, Niš*

# „IO" — COMMODORE



Još jedna igra koja veoma podseća na Nemesis. Cilj je, naravno, stići živ do kraja nivoa, i skupiti sve više poena. Neprijatelji su veoma raznovrsni, a na kraju nivoa posebno će vas namučiti jedan koji je od vašeg broda veći bar četiri puta.

1. NIVO U ovom nivou ključna stvar su meteoriti. Oni vam mogu dvojako poslužiti u napredovanju ka cilju: kao aktra bacači vatre, a takođe i kao energetska polja kadra da vas zaštiti od skoro svih neprijatelja (zmija, lasera itd). Veoma dobra stvar je što kad metak ide na vaš brod a pogodi meteorit, nećete izgubiti život nego će samo jedan meteorit nestati. Taktika kojom ćete uništiti sva vanzemaljce je sledeća: pomerite se brodom na krajnju levu stranu ekrana tako da vam meteoriti služe kao zaklon, a u isto vreme i pucajte. Ovde stvarno treba imati izdržljiv džojstik, jer morate neprekidno pucati sve do kraja nivoa gde vas čeka ogromni svemirski brod. Njega uništavate deo po deo. Koji ćete deo uništiti prvi, nije važno. Važno je da upamtite da na ovom nivou nema stajanja. Morate biti uvek u pokretu i što je najvažnije pucati. Posle toga selite se na nivo 2.

2. NIVO Ovde će vam velike probleme praviti laserski meci koji dolaze iz bilo ka. Ne pomerajte se unutra previše brzo, jer će se desiti da vas laser pogodi. Takođe se ne spuštajte ni previše blizu zemlji, jer vam se može dogoditi isto. Na ovom nivou ćete takođe naići na jednog gadnog protivnika, bacač semena (koje je za vas naravno smrtonosno). Još je nezgodnije što on baca semenje u nekoliko različitih visina pa je preporučljivo skloniti se dole, a onda ga uništiti jednim projektilom. Pri kretanju nadole treba se čuvati već pominjanih lasera. Posle ovoga iskušenje doći će novo: jaja „malih zelenih". Idite što više u levu stranu i probajte da ih izbegnete što više možete. Pazite kako se ovde krećete napred, zato što na vas sve vreme motri jedna ogromna zmija. Kada uđete u pećinu punu laser-bijki idite sasvim levo i uništite neprijateljske brodove koji su se poredali u liniju. Sada se pomerite u bilo koju stranu i izbegnite vatru. Naravno, u obzir uzmite i lasere koji dolaze odozdo. Posle će ponovo doći jaja, ali ovoga puta ih likvidirajte inteligentnim bombama. E, sada dolazi ono najstrašnije: ogromna lobanja koja iz očiju baca vatru. I ovde važi sistem koji smo primenili na prvom nivou: pucajte i krećite se neprestano, ali najviše pucajte. Prosto će vam biti žao kada uništite onako lepo nacrtanu mrtvačku glavu.

3. NIVO Na ovom nivou svemirske kamikaze će vam zagorčavati život. Njihova putanja leta zavisi od vaše, pa za njih nema sigurnog leka. Zato pucajte brže od njih, jer u protivnom ... zna se. Velike glavobolje osim kamikaza, zadavaće vam svemirski mehurići. Najvažnije je da ih pogodite čim ih vidite, jer ćete ih veoma teško izbeći. Međutim, oni ni mrtvi nisu bezopasni, jer za sobom ostavljaju dijamante, koji su za vas takođe opasni. Pošto su dijamanti neuništivi, najpametnije je zaobilaziti ih. Pazite na ogromnu zmiju koja će vas iznenada napasti. Nju međutim nije teško zaobići ako ostanete u sredini ekrana, jer se ona kreće ukrug, a dotle vi pucajte bez prestanka i eliminišite je. Na kraju nivoa će vas čekati ogromni laser ugrađen u stenu. Ako sa sobom imate neki meteorit, to će vam olakšati posao, u protivnom, držite se instrukcija. Prvo treba uništiti generatore ispod i iznad lasera. Idite prvo gore do zelenog generatora i u njega ispalite pet metaka. Ni slučajno više, jer bi to moglo da dovede do punjenja lasera ogromnom energijom, a ovo opet do neželjenih posledica. Isti posao uradite i na donjem generatoru, a onda krenite na veliki laser. Pucajte u pauzi između dva metka koja on ispali. Posle nekog vremena prići ćete na zadnji nivo.

4. NIVO Upozorenje: na ovom nivou ima puno trikova. Pazite se neprijateljskih brodova koji čim pređete određenu liniju kreću za vama ogromnom brzinom. Trebaće vam puno veštine da pređete ovaj nivo, koji pored svega sadrži i zmije i brodove kamikaza. Na kraju nivoa dolazite do ogromnog fetusa, koga treba uništiti. Čim mu priđete, okružiće vas neprijateljski brodovi. Najbolje je da se sklonite ustranu i sve ih uništite. Sada su na redu vrata koja čuvaju fetus od metaka. Kada i njih uništite, fetus će pokušati da se zaštiti, a vi gađajte u oko!

To bi bio čitavo uputstvo za ovu fenomenalnu igru. Pošto je skrolujuća, mapa je nepotrebna, ali je pouk i te kako dobrodošao. Potrudiću se da ga nađem, a onda ga vi potražite u poukovoj mreži.

Vladimir Janković

# ANGLE BALL

Dabome da se iza ovog naslova krije simulacija bilijara. Mnogo toga biće jasno kada pogledate ekran. Možete da igrate na tri nivoa. To su početnički, amaterski ili profesionalni.

Ekran je prepun podataka o udarcima, promašajima, jačini i sličnom, posebno je zanimljivo igrati protiv prijatelja.

Da biste dobili, potreban je matematički način mišljenja.

*Svetozar Risteski, Ohrid*

Nekad bilo

# JET SET WILLY

Ako ijedna igra zaslužuje da se pojavi u ovoj rubrici, onda je to ova. Pojavila se 1984. godine i napravila haos. U to vreme sam sa „Judo" provodio uz „Cookie", „3d tanx", „Android 1" i sl. i onda došao do Vilija. Ne sećam se da li sam bilo koju igru igrao bar upola koliko ovu. Ideja da se igrač kreće kroz sobe nazavisne od toka igre bila je genijalna. Za kratko vreme stvorila se nova vrsta igara, tzv. „platformske igre". Vili je doživeo još tri nastavka, ali nijedan od originalnih autora. Povećavan je ionako veliki broj soba (60). Vili je hodao i skakao brže (JSW II), išao u grad (JSW III), kupovao novi šešir (JSWIV) i još mnogo toga, ali original nikada nije dostignut. Svaka nova soba na koju bih naišao bila je propraćena sa „Joj, šta je ovo?" ili „Koliko ih još ima?", a kada sam najzad saznao da ih je 90 (do tada sam bio otkrio 58) moj „spectrum" je radio i po 10 sati dnevno. Pronalaženja dva poslednja sobe bila je propraćena neartikulisanim izlivima oduševljenja. Verujem da među vama koji ovo čitate ima dosta onih koji su se prepoznali u gornjim redovima.

Bilo kako bilo, desetine programera počelo je da piše platformske igre i smišlja svoje junake, ali ih je malo uspelo u tome („Pyjamarama", „Dynamite Dan" i svaki takav program poređen je sa JSW). Kasnije je većina ipak shvatila da je zakasnila i prestali su da se trude da uđu u legendu. Mesto je bilo popunjeno.

*Aleksandar Pantić, Beograd*

## POUKOVA MREŽA

„Spektrum" — Vladimir Janković

**FIREFLY** POKE 38720,201 vam omogućava igru bez neprijatelja, dok vam POKE 48453,183 daje bezbroj života.

**ISALL II** Za neograničeni broj bombi otkucajte POKE 43366,N (gde je N broj bombi 0—255). Za neograničeno mnogo bombi otkucajte POKE 38920,0. Ako hoćete neograničen broj vatrenih lopti otkucajte POKE 43394,N (n=0-256). Hoćete li da krenete sa nekog višeg nivoa, otkucajte 43410,N (N je broj nivoa). Za neograničen broj života pomoći će POKE 45392,0. Međutim, sa ovim poukom niko vam neće moći ništa: POKE 48000,201

**IKARI WARRIORS** POKE 39273,0 je bezbroj života! Za neograničen broj bombi POKE 40076,183. Municija POKE 39617,183. Sa ovim poukom vreme vam se neće smanjivati: POKE 40818,0, a sa ovim nećete imati problema sa tenkovima: POKE 62384,190

**KARNOV** POKE 32856,286 — bezbroj života.

**MEGA APOCALYPSE** POKE 23578,0

**NORTH STAR** Za bezbroj života POKE 48371,190. Za igru bez neprijatelja: POKE 48283,190. Ako vam nedostatak oružja stvara probleme, otkucajte POKE 43680,0

**ROLAROUND** POKE 30900,0—bezbroj života.

**ROLLING THUNDER** Za bezbroj života potrebno je uneti POKE 39792,0. U slučaju da vam se energija prebrzo smanjuje služi POKE 40013,0. Vreme će biti neograničeno sa POKE 38908,0. Pripremili smo vam i dva iznenađenja. Otkucajte POKE 40319,0 ili POKE 43387,0 i ... videćete.

**SIDE ARMS** POKE 29411,127 — bezbroj života.

**3D STAR FIGHTER** POKE 35714,183 — bezbroj života.

**ARKANOID II** POKE 33055,255,POKE 39969,0. U slučaju da posedujete verziju 128 K pouk je POKE 37586,0.

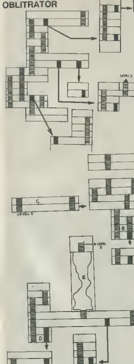
**BLACK LAMP** Za bezbroj života morate uneti dva pouka: POKE 33606,127, POKE 34467,127

**BLIND PANIC** Neograničenu energiju dobićete poukom POKE 40586,201. Ključevi vam više neće predstavljati probleme sa POKE 38688,195. Ako vas mrzi da počinjete od prvog nivoa, POKE 41013,N će vas odvesti u bilo koji nivo (N-nivo)

**CROSFIRE** POKE 33848,9—besmrtnost.

**CYBERNOID** U jednom od prošlih „Računara" bila je objavljena mapa, a sada vam dajemo kompletan sistem za varanje. POKE 36967,0 će vas učiniti besmrtnim. Ako vam oružje stalno nestaje, POKE 31798,0 će to srediti. Unesite POKE 38817,195 i niko vam neće moći ništa. Za kraj, ako vam CYBER MACE pravi problem tu je i POKE 27210,0. Do skorog viđenja.

# OBLITRATOR




Vaš cilj u ovoj igri je da u ulozi Draka uništite neprijateljski brod koji namerava da razori vašu rodnu planetu. Na početku igre Drak će biti teleportovan na jednu od lokacija na brodu koju će kompjuter odrediti randomom. Takođe, Drak će na početku biti naoružan samo pištoljem, naravno laserskim. Neprijateljski brod sastoji se iz nekoliko sektora (nivoa) koji su povezani liftovima. U vašoj misiji pomagaće vam zaštitno odelo kao i bezbroj korisnih stvari koje ćete nalaziti uz put. Draka takođe možete kontrolisati pomoću ikona. Veliku nevolju zadavaće vam platforme, sa kojih ne smete pasti, jer će dovesti do kraja igre. Kada ulazite u novi ekran držite kursor između ikone za napad i ikone za od-



## LEGENDA

SVEMIR/PRAZAN PROSTOR

LIFT -

IZLAZ IZ LIFTA -

GRAĐEVINE -

branu tako da nećete morati da odvajate oči od ekrana na kome vrve neprijatelji. Pazite se rapidne paljbe neprijatelja, jer će vam biti teško da je izbegnete, što može dovesti do Drakove smrti. Aktivirajte mašinu Plasma Engine Drive posle svih drugih predmeta koji su se približili šatlu. Ikona za vraćanje je krajnje desna, a mesto ćete naći sami. Ovo je neophodno za kompletiranje igre.

Takođe za lakše igranje dobijate i mapu. Na mestima obeleženim sa A,B,C D,E sigurno ćete imati problema. Evo nekih objašnjenja.

A) Na ovom ekranu nalaze se četiri lasera. Trebaće vam dosta vežbe da ih zaobiđete bez gubljenja energije.

B) Ovde se nalazi nevidljivo zaštitno polje. Da biste prošli morate upotrebiti mali trik. Pucajte laserskim pištoljem u kutiju na levoj strani ekrana — i slobodno prođite.

C) Ovde će vam put zagraditi veliki zid koga je nemoguće rasturiti običnim laserom. Međutim, pronađite negde Razarač (big blaster) i pucajte u zid — za slobodan put.

D) Na ovom ekranu vam se može desiti da uletite u zamku. Naime, zarobiće vas dva lasera, a ako se to desi teško ćete se oslobodili.

E) Upozorenje. U ovom sektoru možete upotrebljavati samo ikone za kretanje (levo, desno, gore, dole). Možda će vam se na početku učiniti komplikovano, ali u suštini (sa malo vežbe) veoma je lako.

Eto to su bili glavni saveti kako pomoći Draku da po X-ti put spasi zemlju. Sa mapom i malo vežbe to će vam poći za rukom.

*Vladimir Janković*

Nekad bilo

# JUGERNAUT

Program nas stavlja u ulogu vozača kamiona koji po gradu pronalazi skladišta i robu u određenoj količini prevozi kući. Postoje 4 vrste tereta: Timber (drvo), Koli (ugljen), Oil (ulje) i Fruit (voće). Po učitavanju dobija se meni sa ikonama. Kada definišemo tastere, odredimo veličinu kamiona i uvježbamo kretanje te krećemo u igru. Kompjuter nam saopštava koliko pojedine robe treba prevesti i pokazuje mapu grada, gdje samo uočavamo startnu poziciju (H). Nakon toga se prikazuju nove ikone. Ikona Phone (telefon) vrlo je važna za pronalaženje pojedinih skladišta. Kada pored ceste uočimo telefonsku govornicu, zaustavimo kamion i pozovemo ikonu. Nakon zvonjenja telefona prikazuju se pojedini proizvodi a njihovim pozivanjem na mapi grada doznajemo položaj, to se odnosi i na benzinske crpke. Posebnu pažnju obratite na pokazivače goriva i oštećenja, jer će igra brzo završiti. Ako prevezete svih 40 tona, kompjuter vam čestita i prikazuje tabelu sa količinom prevezene robe i odbitke za gorivo i oštećenja. A zatim ponovo za volan — i sve ispočetka.

*Milan Vujasić, Gora — Petrinja*

# BREAKER

## „Komodor"

U verziji koju ja posjedujem, na početku igre birate da li želite besmrtnost ili normalnu igru i da li želite mijenjati nivoe. Ako želite mijenjati nivoe, dovoljno je da u toku igre pritisnete + za prelazak u sljedeći nivo ili — za vraćanje u prethodni nivo. Igra se sastoji iz 64 nivoa. Možete čak praviti sopstvene nivoe (tako što ćete u osnovnom meniju strelicu postaviti na EDITOR).

Vaš zadatak u ovoj igri je da pomoću palice i loptice razbijate zid. Pri tome će iz zida padati srca. Srca vam donose razne nagrade i olakšavaju igru, ali njihovo hvatanje nije obavezno. Međutim, ako dvaput uhvatite isto srce, nagrada koju vam je ono donijelo se oduzima. U igri postoji mnogo srca sa raznim nagradama, ali ću ovdje opisati samo ona sa kojima se u igri najčešće susreće.

Crveno srce sa oznakom C. Kada ga uhvatite vaša palica dobija žutu boju i postaje ljepljiva, pa se loptica svaki put zaustavlja na njoj. Za ponovno pokretanje loptice pritisnite FIRE.

Crveno srce sa oznakom B dobijate nagradu život.

Crveno srce sa oznakom G. vaša loptica se znatno usporava.

Zeleno srce sa oznakom E vaša palica se povećava.

Plavo srce sa oznakom B. u igru se ubacuje još jedna loptica.

Plavo srce sa oznakom L. kada ga uhvatite na krajevima palice se pojavljuju produžeci kojima možete razbijati cigle pritiskom na FIRE.

Plavo srce sa oznakom P sada palica postaje plava, a umjesto nje upravljate lopticom (morate paziti da poslije svakog hica lopticu spustite baš na palicu).

*Goran Đukanović, Tesić*

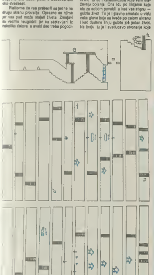
# ICE PALACE



Na scenario igre nećemo trošiti puno reči. Sve je već viđeno, samo što se ovde radnja događa u Ledenom zamku. Najvažnije je znati da su po zamku rasute razna vrsta oružja. Najefikasniji je tanki mač. Veliku monstrumu eliminišete tako što ih mačem pogodite oko deset puta, a buzdovanom oko dvadeset.

Platforme će vas prebaciti sa jedne na drugu stranu provalije. Oprezno sa njima jer vas pad može stajati života. Zmajevi su veoma neugodni jer su sastavljeni iz nekoliko delova i svaki deo treba pogoditi po nekoliko puta. Ptice su najgore od svih neprijatelja i za njih nema pouzdanog leka. Pouk je već ubačen (verzija C 64), pa uz mapu ova igra neće predstavljati veliki problem.

*Vladimir Janković*



# MANIAX

Evo još jedna igra koja pleni svojom jednostavnošću. U ulozi ste malog romboida koji treba da obiđe 75% ekrana. Prva novost je što pozadinu sačinjavaju razne kulturna obeležja. Na svakom je po jedno obeležje (nivoa ima pet) koje se postepeno pojavljuje kako vi zatvarate ekran. Naravno, tu su i karakondžule koje vam otežavaju bojanje. One idu po linijama koje ste za sobom povukli, a kad vas stignu — gubite život. Tu je i glavno smetalo u vidu neke glave koja se kreće po celom ekranu i kad dodirne liniju gubite još jedan život. Na kraju tu je i svetlucavo stvorenje koje je veoma brzo i pojavljuje se samo kada dugo oklevate. Kada linijom zatvorite neki deo obojiće se onaj u koma se ne nalazi glavno smetalo. Vaš zadatak je da dobijete slike Koloseuma na prvom nivou, Akropolja na drugom, Ajfelovog tornja na trećem, asteških piramida na četvrtom i Kipa slobode na petom nivou. Na svakom nivou skupljajte i slova, jer ona daju nazive gradova u kojima se obeležja nalaze: Rima, Atina, Pariza, Meksiko Sitija, Njujorka. Takođe će se povremeno pojavljivati trouglovi, srca, znakovi pitanja, koji vam mogu davati mnoga poboljšanja ali i pogoršanja (brzina, odnos poena, zaustavljanje vašeg romboida, usladavanje...). Kad završite neki nivo, možete dobiti novi život ili bonus poena.

*Mićić Dragan i Nikolić Darko, Beograd*

# DARK EMPIRE

Nakon dužeg vremena evo jedne dobre strateške igre. Cilj vam je da uništite sve neprijateljske jedinice i osvojite protivničke gradove. Od snaga na kopnu imate samo pešadiju, ali se to nadoknađuje na moru gde možete koristiti izviđački brod, razarač, podmornicu, transporter.

Na početku startujete sa jednim gradom, pešadijskom jedinicom i transporterom. Da bi dobili još jedinica stavite kursor na grad i pritisnite pucanje. Sada pomeranjem palice dole ili gore odaberite koje vam jedinice treba, pritisnite pucanje i posle nekog vremena željena jedinica će se pojaviti u okolini grada. Neka pogodnost možete dobiti ako pritisnete F7, a potom u novom meniju odaberete OPTIONS. Sada koristite tabelu koja je objavljena u RAČUNARIMA 40. Jedinice koje služe za kretanje po moru (razarači, podmornice i sl.) može da proizvedi samo grad koji se nalazi na obali, a pešadiju bilo koji grad.

Jedinice možete zaustaviti (HALT), narediti im da idu u pravcu koji ćete odabrati pomoću brojava, a možete im i narediti da odu od nekog mesta koje ćete označiti kurzorom. Pešadijske jedinice možete prebacivati preko mora transportnim brodovima, tako što dovedete jedinicu na brod. Kad na mapi dođete do kraja pojavićete se na suprotnoj strani. Ovo treba koristiti da bi se kompjuteru zašlo za leđa jer ponekad zna da postavi svoje jedinice duž obala tako da vi ne možete da iskrcate svoje trupe. Ovo ponekad uradite i vi, sačekajte da dođu neprijateljski transporteri pa ih zatim mirno skinite sa razaračem ili podmornicom. Ako se na putu kojim ide vaša jedinica nađe neprijateljska jedinica ili grad, borba će automatski početi čim se dotaknu i za razliku od TOBRUK-a, trajaće samo nekoliko sekundi. Kada počnete sa igrom primetićete da je samo mali deo mapa oko vaših jedinica vidljiv, dok je sve ostalo jednobojno. Da biste videli šta se tu nalazi morate proći i „osvetliti" taj deo mape. Celu mapu možete videti ako pritisnete F1, a ako hoćete uvećanu sliku mape da biste lakše razabrali jedinice pritisnite SHIFT. Jedina mana je da animacija trenutne pozicije na kaseti zauzima previše mesta. Odlična igra za prave stratege.

*Aleksandar Ristić, Titovo Užice*

## Ever green never green

*Ever green* — *Miklija Kostića*

### SPELLBOND

Oduvek sam volela spoj avanture i blage akcije, i još ako je begat dobrom grafikom i duhovitim scenarijem, onda je to pravi program za mene.

*Never green* — *Dušana Katilovića*

### DRUID II

Kako prvi deo nije bio loš i malo mu je nedostajalo da bude i vrhunski, mislio sam da će sitne greške biti ispravljene, ali sam se prevario. Grafika je više nego očajna, animacija vrlo loša, a ponekad se druidk jedva vidi. Pa kako ovo igrati?

*Ever green Nina i* — *Gorana Petreka*

### INTERNATIONAL KARATE

Ovo je najbolja simulacija karatea. Grafika i animacija su za pet plus. Udaraca i pokreta ima mnogo. Ovu igru nemojte propustiti!

*Never green* — *Aleksandra Milićevića*

### CEPELIN

U ulozi ste cepelina, letite kroz neke podzemne hodnike, i pucate u svim smerovima. Želela mi ništa nije jasno. Grafika i zvuk su očajni. Užasno!

*Ever green* — *Predraga Zerda*

### RENEGADE

Dobra grafika, puno zvučnih efekata i nadasve dobra ideja čine da ova igra zaista zaslužuje mesto među najboljima. Držala me mesecima!

# MIRISI IZ KUHINJE

*Voleo bih da me razvoj događaja demantuje*

*Imam utisak, naime da jesenje igre ni izdaleka neće biti „za otkidanje" Proizvođači uglavnom operišu onime koje su već uveliko lansirane nastojeći da iz njih iscede što više mogu Neke prebacuju sa kaseta na diskete neke grupišu za prodaju u paketu, neke opet prerađuju manje ili više korenito za prodaju pod novim imenima i to je mešlena sve*

*Nove igre ponekad su nastavci starih (BARBARIAN II na primer) ili nova verzija stara teme (kao što će biti igra o Supermanu koja zasad još ni naslova nema. Samo se zna da neće biti podgrejani SUPERMAN iz 1985, i da se pravi za „Komodor" vulgaris i „amigu" Drugi autori druga firma, drugi pristup ali isti lik.)*

*„Kad se kaže zna se šta je" govorio je nekad između ostalog, znameniti Beograđanin Pera Zvezdaš. Na tu njegovu maksimu podsetila me je igra s naslovom TIGER ROAD priznajemo, ovde se odmah zna šta je Daleki istok borilačke veštine, nindže zmajevi japanski likovi u kineskom ambijentu sve je jasno kao dan*

*Ako nemaš ničeg novog drži se starog i proverenog, i tako je nikio još nekoliko pucačkih igara LOCK ON i ARMALITE na primer LOCK ON je možda teži ali je ARMALITE ako ništa drugo grafički bolji Kudikamo bolji,*

*Što zbog Olimpijskih igara što po merciji spremljene je i pristojna količina sportskih simulacija SUMMER OLYMPIAD EDDIE EDWARDS SUPER SKI STREET SPORTS SOCCER itd. Sprema se čak i kompjuterska verzija vanredno popularnog Bi-bi-sijevog TV kviza A QUESTION OF SPORT Možda najviše obećava STREET SPORTS SOCCER zato što se u njoj fudbal ne igra na stadionu nego na uličnom terenu koji je ovakav kakav je to jest nalik onima na kojima smo i mi šutirali loptu.*

*Pojen sam od kokoške nogu ali ne sasvim stoga ću samo u pola glase reći da je gotovo sasvim skuvana jedna izvanredna domaće igra. Još se ne zna ko će joj biti izdavač ali se zna da je izuzetno teška da ima okruglo sto nivoa, i da je strogo cerebralna Grafički je jednostavna, ali upečatljiva Prve konkretnije vesti o njoj čućete u ČIPU I SEDAM JARIĆA i čitati u RAZBARUŠENIM SPRAJTOVIMA.* **V SL**

## Poukova mreža

### Amstrad

**PHANTOM CLUB** — energija
10 OPENOUT "TPL" MEMORY &35AA
20 LOAD "!pc-1"
30 CALL &4000
40 LOAD "!pc-2"
50 POKE &62CE &B7
60 CALL &35AB

*Domagoj Marić*

## Sveže učitano
### Spektrum

## SABOTAGE
### SABOTAŽA

*Kako su se vanzemaljci ostrvili na dragu nam planetu, to se prosto ne da zamisliti Opet su se neki pojavili iz bestragije da nas sve ojade i opet ste vi jedina šansa čovečanstva.*

*Dakle letite i pucate pucate i letite*

*ali nije cela priča u tome Da biste uništili tuđinske instalacije morate se u svakom sektoru (ima ih ukupno 8) domoći jednog dela šema, tek kad sastavite delove u celinu možete se nečemu nadati Delove ćete dobijati od tuđinovih sabotera (u svakom sektoru čuči po jedan) Do sabotera dolazite kad prođete kroz lavirint u kome vas napadaju ptice kamikaze Do*



*lavirinta dolazite kad nekolikim pogocima uništite matični brod A kako dolazite do matičnog broda?*

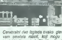
*E tu je čvor Do njega dolazite — ako uopšte dođete — probijajući se kroz gomila smetala Pošto su sva do jednog opasna po život (veš) moraćete ih ili upucavati ili izbegavati Kako ćete to izvesti, vaša je stvar*

*Atari*

## QUADRALIEN
### ČETVOROTUĐIN

*Cerebralna igra lukavo sakrivena u arkadnoj Nešto kao Karađorđeve šnicle ili zagrebački odrezak*

*Arkadnog orgijanje ima koliko hoćete orbitalnu elektranu treba da očistite od radioaktivnog materijala. Za taj posao ćete odabrati dva androida od šest koliko vam stoji na raspolaganju. Oni treba da prođu četiri nivoa i u svakom po šest lokacija, pri čemu će nailaziti na zamke i smetala, biće spektakularne pucnjave*



*Cerebralni deo izgleda ovako glavna su vam smetala roboti, koji mogu biti atraktori (A) ili repeleri (R) Kao što im i imena kazuju atraktori se međusobno privlače a repeleri odbijaju Možete svakog muvati po prostoriji i uništiti laserom ali kako će se kretati oni pored kojih tej prolazi? I kako će delje uticati jedni na druge? To je ono što morate prokljuviti, razmišljajući strogo logički.*

*Naravno svaki je nivo zagonetka za sebe — što znači, između ostalog da će vam svake moždane vijuge biti engažovane.*

*Amstrad CPC*

## NORTHSTAR
### SEVERNJAČA

*Zapravo, nije zvezda nego svemirska stanica.*

*Na njoj se sjajno proizvodile hrana za prenaseljenu i gladnu Zemlju a onda su je zauzeli zli vanzemaljci Vaše je da ih likvidirate i da reaktivirate sistem za održavanje života (ljudskog) Vi možete opstati i bez sistema jer ste delimično robot. usađeno vam je nešto robotskih elemenata da biste bili efikasniji Među te elemente spada i ubitačne robotske ruke veoma korisne kao oružje u borbi protiv uzurpatora.*

*Ne moram ni reći da vam se arsenal može bitno povećati Ima i drugog oružja, još ubojitijeg nalazićete ga uz put, dok obilazite stanicu*

*A biće vam i te kako potrebno, jer ćete se sukobljavati s neprijateljima svih boja*



*Vremenom ćete ustanoviti koje je oružje najpogodnije za koju priliku*

*Reaktiviranje sisteme nije toliko opasno — radi se mahom o izvršavanju radnji po pravom redosledu — koliko zamršeno ali to ste svakako i očekivali znajući da u dobroj igri nikad ne teku med i mleko*

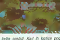
*Amiga*

## JINKS
### PROVALE

*Leta ružičasti prasići i plišani medvedići, rolinaju planete vrie se burgije u formacijama nalaću cigle zašto PROVALE pitate se zašto se ovo ne zove DELIRIJUM?*

*Pitanje ostaje bez odgovora. Naslov je ionako nevažen.*

*Važno je ovo u ove elemente delirijuma — ali ne samo u njih — pogađa loptica koju udarate trouganonom maškom Tačnije loptice koju udarate bilo kojom stranom te maške zavisno od toge u kom je pravcu želite poslati Kad ih loptica pogodi elementi groktu (prased) cića (meda) eksplodiraju (cigle) ili prosto-naprosto daju bonus poena*



*Lopticu morate juriti po ekranu, i tom prilikom ćete naletati na svašta od nekih predmeta maška će se smanjivati od nekih se oporavljati od nekih opet, ulazili u bonus nivo, itd*

*Ako kažete da sve ovo liči na BREAKOUT i njegove klonove bićete u pravu Ako kažete da vam liči i na flipere bićete opet u pravu*

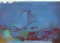
*No, jamačno nećete reći da je dosadno.*

*Amiga*

## RETURN TO ATLANTIS
### POVRATAK U ATLANTIDU

*Još jednom ne verujte naslovu Atlantida je tek na kraju igre, Atlantida je vaša četrnaesta misija. Prethodi joj razume se, tršta muka i nevolja*

*Kao prvo, birate kakvi ćete biti nudi vam se šest ličnih osobina a vi treba da ih po svom nahođenju složite u opadajućem nizu (tj najvažniju na prvo mesto i tako redom) Potom odlazite u kafanu i razgovarate sa gostima (ima ih pet) ne biste li čuli neki konstan podatak. Ako negda ne pomaže uljudnost sagovorniku možete zapretiti — ili ponuditi mito kako precente*



*Pokupivši sve podatke sedate u komandni brod i krećete u akciju uz pomoć mapa sveta. Kad stignete na odredište izlazite iz broda ronite do dna — i tamo, naravno berete kožu na šiljak, jer očekuju vas svakojake podvodna smetala. Na sreću, imate laser — i još važnije mini-podmornicu koju možete programirati tako da vam bude izvidnik i lokator traženih predmeta. Šta pokupite to šaljete u komandni brod — a često ćete i sami morati tamo, da obnovite zalihe kiseonika i drugih potrepština.*

*Ako uspete vaše se osobine unapređuju i određuje vam se sledeća misija, sve do Atlantide*

*Amiga*

## RETURN TO GENESIS
### POVRATAK POSTANJU



*Scenario je Lestrage dugačak i bombastičan brljenini naučnici besmrtni klonovi let u budućnost, zli tuđini vrhunska tehnologija i tome slično*

*Naravno u pitanju je obična pucačka igra*

*Povrh pucanja i vrdanja, imate i jedan human posao treba da oslobađate naučnike (ili, ako hoćete klonove naučnika), kojih ima sve u svemu šest stotina. Jedni će imati da vam kažu ponešto korisno; drugi imaju podatke vama nepotrebne; ali unapred ih ne možete razlikovati.*

*Korisni podaci liču se dopunskog oružja i opreme ali ništa nije večno neke olakšice traju samo po desetak sekundi. Neke opet, prestaju tek kad izgubite tekući život*

*Kad zađete dublje u igru otkrićete da vas nailetanje na izvesne predmete ne ubija nego samo odbacuje unatrag. Tada se može lako desiti da vas sustigne neprijateljska vatra što će vas podsetiti na onu narodnu „Bežale ribe s lovanja pa pale na plotnu."*

*Ostalo je pucnjava.*

# PC810

**ELEKTRONIKA**
**TELEKOMUNIKACIJE**
**INFORMATIKA**



PC 810 je reprezentativni model nove tročlane UNIS — NCR serije personalnih računara

PC 710
PC 810
PC 916

Odlikuje se tzv Incremental Workstation Architecture koja omogućava korištenje najnovija tehnologija pojednostavljuje servisiranje i pruža mogućnost proširivanja početnih do veoma moćnih konfiguracija

Uzimajući sve prednosti dizajna integrisanih ploča (VLSI) i visoku tehnološku opremljenost PC 810 nudi pojačane AT-učinka Šest AT-kompatibilnih slotova za proširenja pružaju mogućnost izbora različite grafike i konfigurisanja različitih memorijskih medija i čine PC 810 snažnim i dovoljno fleksibilnim za većinu korisničkih zahtjeva
U PC 810 je na osnovnoj procesorskoj ploči ugrađen 16-bitni INTEL 80286 mikroprocesor koji funkcioniše dvojnom brzinom od 6 ili 10 MHz Na istoj ploči nalaze se i podnožja za ugradnju aritmetičkog koprocesora 80287 ROM-čipa za zaštitu za korisnika specifičnog softvera kao i 640 KB RAM memorije Korištenjem ploča za proširenje može se RAM memorija proširiti i do 16 MB

### OPERATIVNI SISTEM

— PC kompatibilan NCR-DOS omogućava korištenje široke palete profesionalnog softvera
— XENIX (opcionalno) je multi-user multi-tasking operativni sistem opšte namjene Prilagođen je personalnim računarima čija je osnova INTEL-ov mikroprocesor 80286
Kod PC 810 je predviđena i mogućnost ugradnje INTEL 80386 mikroprocesora čime ovaj model prerasta u najsavršeniji iz nove serije — PC 916

### KOMUNIKACIONE MOGUĆNOSTI

PC 810 podržava sve standardne komunikacione protokole i može se vezati u mrežu sa proizvodima svih u svijetu priznatih proizvođača računarske opreme

### TABELA SA OSNOVNIM KARAKTERISTIKAMA

| | |
|---|---|
| CENTRALNI PROCESOR | INTEL 80286 (opcionalno 80386) |
| KOPROCESOR | INTEL 80287 |
| BRZINA RADA | 6/10 MHZ |
| GLAVNA MEMORIJA | 640 KB — 16 MB |
| DISKETNA JEDINICA 5 1/4" | 1 2 MB |
| DISKETNA JEDINICA 3,5" | 2×720 KB ili 1 44 MB |
| WINCHESTER DISKOVI | 2×20/30/44/70/115 MB |
| STREAMER TRAKA | 40 ili 60 MB |
| GRAFIČKE KARTICE | CGA, EGA i VGA |
| EKRANI | 15 Monohrom, 14 Kolor |
| REZOLUCIJA EKRANA | MONO 640×(200×2)<br>KOLOR 640×200 |
| INTERFEJSI | RS 232, CENTRONICS |
| RADNI USLOVI — Temperatura | 10—32°C |
| — Rel vlažnost | 20—80% |
| OPERATIVNI SISTEM | NCR DOS, XENIX, predviđen OS/2 |
| KOMUNIKACIJE — Protokoli | TTY HDLC SDLC<br>3270/3770 BSC, 2780/3780 BSC |
| — Mreže | SNA, X.25, TOKEN RING |
| KOMPATIBILNOST | AT i PS/2 KOMPATIBILAN |



OOUR „MARKETING"
Sektor plasmana opreme za AOP

71000 Sarajevo
Trćanska 7

Tel 071/205-455
Telex 41576/52